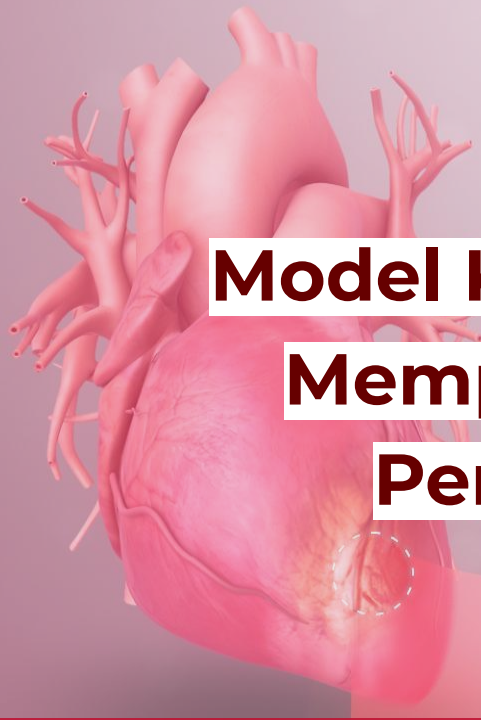
Data Science
Department of Mathematics
Faculty of Mathematics and Natural Sciences
University of Indonesia



Model Klasifikasi ANN dalam Memprediksi Keberadaan Penyakit Jantung

Argama Vanesa
2306207676

Millah Nafisah
2306244021

Muhamad Erik S
2306203620

Raditya Fauzan
2306244186

Kelompok 6

Apa itu Penyakit Jantung?

Penyakit jantung atau **penyakit kardiovaskular** adalah gangguan pada **jantung** dan **pembuluh darah**. Salah satu bentuk paling umum adalah **penyakit jantung koroner**, yaitu **penyempitan pembuluh darah** yang mensuplai jantung akibat **penumpukan plak** (**aterosklerosis**). Jika tidak ditangani, dapat menyebabkan **serangan jantung** atau **gagal jantung**.

Mengapa hal ini penting?

Penyakit jantung merupakan **penyebab kematian utama** di dunia. Meski sering **dapat dicegah**, penyakit ini berkembang secara **diam-diam** dan tanpa **deteksi dini** dapat menyebabkan **serangan jantung mendadak** menyulitkan **penanganan tepat waktu** dan membebani **perawatan jangka panjang**.

Indikator Penyakit Jantung


Usia


Jenis Kelamin


Nyeri Dada


Tekanan Darah Tinggi

Kolesterol Tinggi


Gula Darah Tinggi


Gangguan Irama Jantung

Detak Jantung Maksimum


Angina


Segmen ST


Penyumbatan Pembuluh Darah


Thalassemia

Tujuan Proyek

Menyelidiki bagaimana indikator **ada / tidak adanya penyakit jantung** dengan **13 fitur** yang tersedia pada dataset dengan melakukan proses **Klasifikasi** berbasis **ANN (Artificial Neural Network).**

Apa itu Klasifikasi?

Klasifikasi adalah salah satu teknik dalam ***machine learning*** atau **statistika** yang digunakan untuk **mengelompokkan data ke dalam kategori atau kelas tertentu** berdasarkan ciri-ciri atau fitur yang dimiliki data tersebut.

Apa itu ANN?

ANN (*Artificial Neural Network*) adalah metode machine learning dengan beberapa lapisan, yakni Input Layer (menerima data dari fitur dataset), Hidden Layer (memproses data), dan Output Layer (menghasilkan prediksi (klasifikasi)). ANN sering digunakan untuk klasifikasi, seperti deteksi spam atau diagnosis medis.

Overview dataset

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
45	1	1	110	264	0	0	132	0	1.2	2	0.0	7.0	1
68	1	4	144	193	1	0	141	0	3.4	2	2.0	7.0	2
57	1	4	130	131	0	0	115	1	1.2	2	1.0	7.0	3
57	0	2	130	236	0	2	174	0	0.0	2	1.0	3.0	1
38	1	3	138	175	0	0	173	0	0.0	1	NaN	3.0	0

Sumber dataset

Dataset ***Heart Disease*** diambil dari UCI Machine Learning Repository menggunakan pustaka **uci repo** yang diinstal melalui perintah **!pip install ucimlrepo.** Lalu di masukan ke *dataframe* sebagai df untuk dilakukan analisis lebih lanjut

Encode variabel kategorik..

Karena semua fitur yang **berupa kategorik** sudah dalam berbentuk *integer* maka tidak perlu di *encode*

Pengecekan missing Value..

```
Kolom dengan missing values:
ca      4
thal    2
dtype: int64
Class distribution for 'num':
```

Terdapat **6 missing value** yang akan di ***handling***

Data tersebut merepresentasikan proporsi data yang hilang **kurang dari 5%**

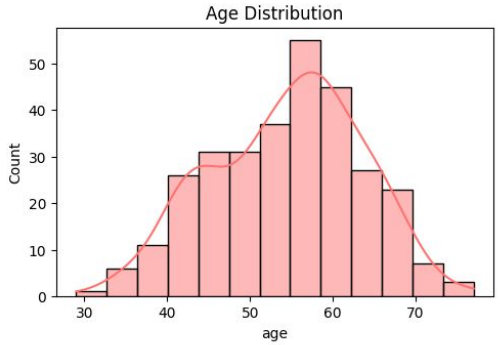
Skala dari tiap data..

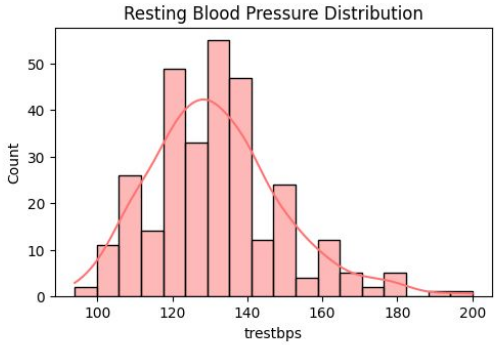
Data memiliki **skala yang bervariasi** pada setiap fitur, sehingga akan dilakukan **proses *scaling*** agar setiap fitur memiliki kontribusi yang seimbang dalam pemodelan

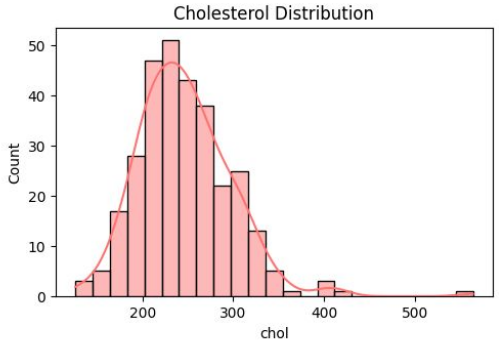
Variabel Dependen	Num	Indikator ada/tidaknya penyakit jantung (0 = tidak, 1–4 = tingkat keparahan)

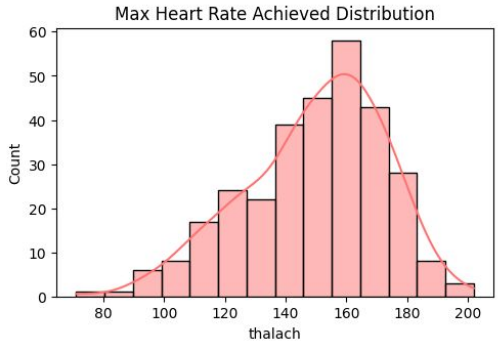
Variabel Independen	Deskripsi	Tipe Data
age	Usia pasien dalam tahun Deskripsi	Continuous Numerical
sex	Jenis kelamin pasien (1 = laki-laki, 0 = perempuan)	Binary (Nominal)
cp	Tipe nyeri dada (1-4)	Categorical (Ordinal)
trestbps	Tekanan darah saat istirahat (mm Hg)	Continuous Numerical
chol	Kadar kolesterol serum (mg/dl)	Continuous Numerical
fbs	Indikator kadar gula darah Puasa >120 mg/dl (1 = ya, 0 = tidak	Binary (Nominal)
restecg	Hasil Elektrokardiogram saat istirahat (kategori 0-2)	Categorical (Ordinal)
thalch	Denyut jantung maksimum yang di capai selama uji beban	Continuous Numerical
exang	Indikasi angina akibat aktivitas fisik (1 = ya, 0 = tidak)	Binary (Nominal)
oldpeak	Penurunan segmen ST akibat latihan fisik	Continuous Numerical
slope	Kemiringan segmen ST saat latihan (kategori 0-2)	Categorical (Ordinal)
ca	Jumlah pembuluh darah utama (0–3)	Discrete Numerical
thal	Hasil tes thalassemia	Categorical (Ordinal)

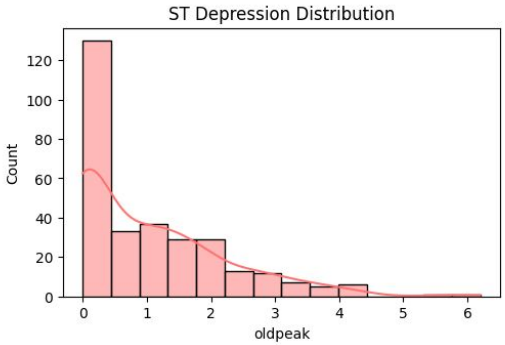
Distribusi Variabel Numerik







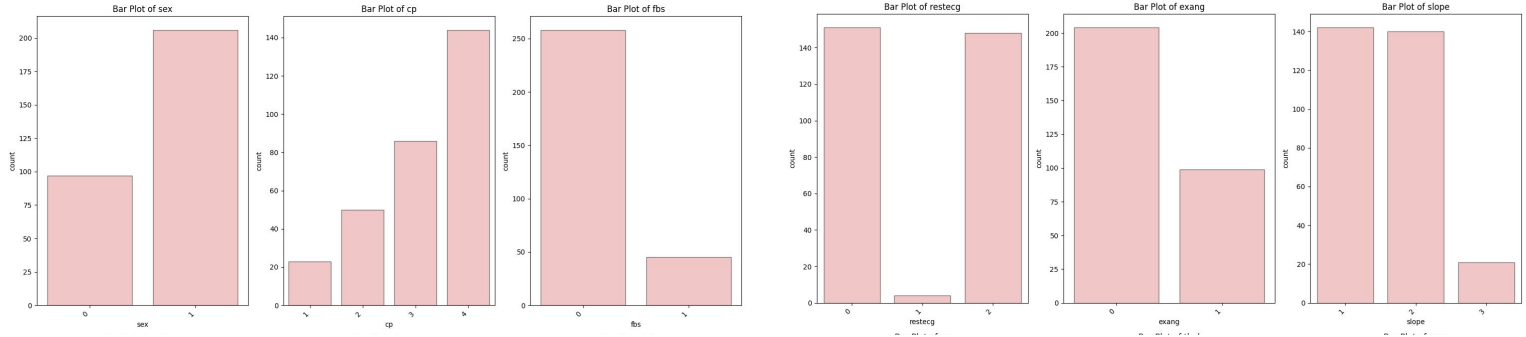




Insight

Visualisasi **histogram** menunjukan bahwa **distribusi data** pada **fitur numerik** pada dataset **tidak seimbang**. Fitur age dan thalach cenderung **miring ke kanan** sedangkan fitur trestbps, chol, dan oldpeak cenderung **miring ke kiri**.

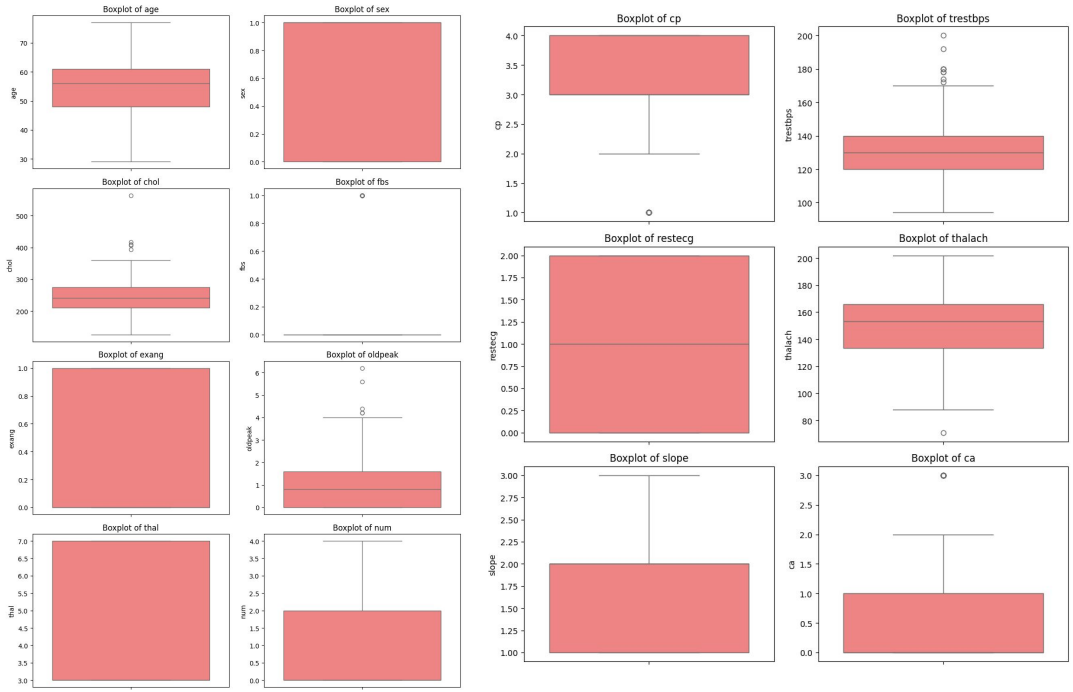
Distribusi Variabel Kategorik



Insight

Visualisasi **bar plot** menunjukan bahwa **distribusi data** pada **fitur kategorik** pada dataset **tidak seimbang** dan terdapat kategori dengan **frekuensi sangat rendah** yang berpotensi menjadi outlier.

BoxPlot Seluruh Variabel



Insight

Visualisasi **box plot** menunjukan bahwa beberapa **fitur numerik** pada dataset **memiliki outlier** yang dapat memengaruhi model.

Handling Missing Values

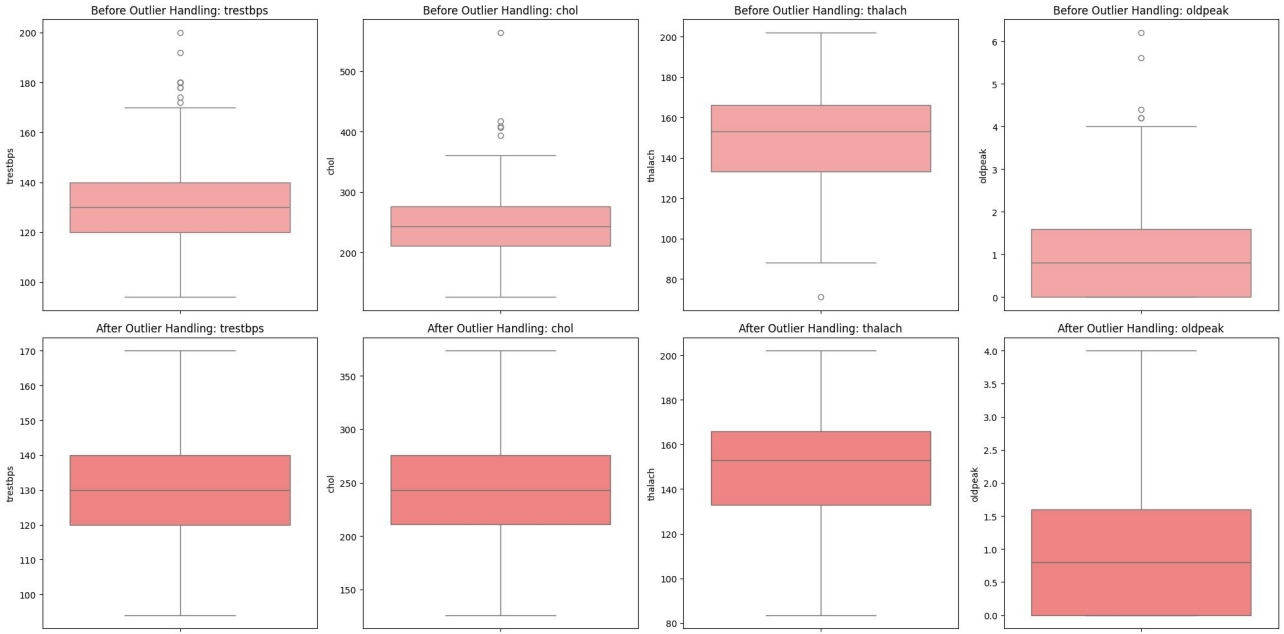
Missing values adalah kondisi ketika suatu data atau atribut dalam dataset **tidak memiliki nilai** yang tercatat **(NaN, None, Null, dsb)**. ***Handling missing values*** adalah proses untuk **menangani data** yang memiliki **nilai hilang** untuk meminimalkan bias, menjaga kualitas data, dan mencegah *error*.

Dataset kami terdiri atas **14 kolom** dan **303 baris** dan **data** yang **hilang** terdapat **6 buah**. Artinya, **proporsi** data hilang **kurang dari 5%**. Oleh sebab itu, kami memutuskan untuk **menghapus baris** yang memiliki data hilang (***deletion method***) dengan **dropna()**.

```
1 print(df.isnull().sum())
```

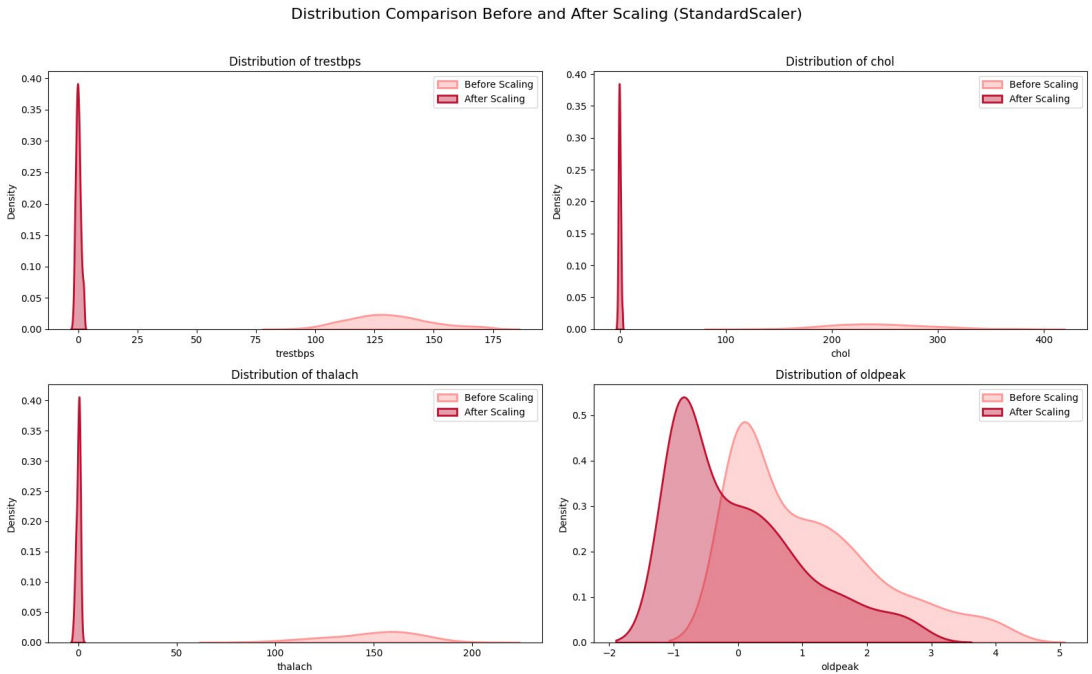
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
num         0
dtype: int64
```

Handling Outliers



Penanganan outlier dilakukan dengan **metode *capping***, yaitu dengan menentukan **kuartil data** dan jangkauan interkuartilnya (*interquartile range*/IQR). Berdasarkan nilai tersebut, ditetapkan batas atas dan batas bawah. Nilai-nilai yang **lebih kecil dari batas bawah** maupun **lebih besar dari batas atas** dianggap sebagai **outlier** dan **digantikan dengan nilai batas bawah** atau **batas atas sesuai posisinya**.

Scaling



Selanjutnya dilakukan ***scaling*** pada **fitur numerik** karena rentang **nilai yang bervariasi**. Kami menggunakan **StandardScaler** agar distribusi data numerik memiliki ***mean* 0** dan **standar deviasi 1**, yang sesuai untuk algoritma yang sensitif terhadap skala fitur.

Cara Kerja StandarScaler

StandarScaler menggunakan **metode normalisasi Z-Score**, yaitu teknik yang mengubah data agar memiliki **distribusi Normal (Gaussian)** dengan rata-rata 0 dan standar deviasi 1.

Langkah-Langkah StandarScaler

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}$$

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Menghitung *Mean* tiap Fitur

Untuk setiap j = 1, 2, .. , 13; hitung **rata-rata (*mean*)** dari **fitur** ke-j

Menghitung Standar Deviasi tiap Fitur

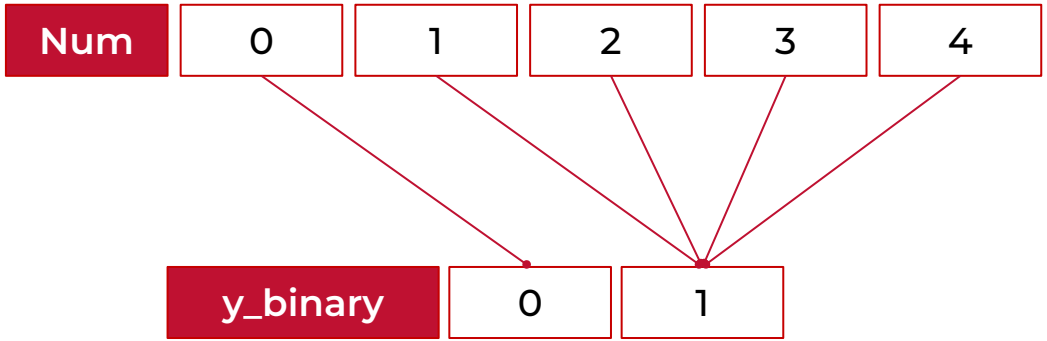
Untuk setiap j = 1, 2, .. , 13; hitung **standar deviasi** dari **fitur** ke-j

Lakukan Normalisasi tiap Data

Untuk **setiap data** pada dataset, lakukan **transformasi Z-Score**.

Membagi Kasus Target

Target pada dataset terdiri dari **lebih dari dua kelas**, sehingga dilakukan dua pendekatan klasifikasi, yaitu **klasifikasi multikelas** dan **klasifikasi biner**. Untuk klasifikasi biner, ditambahkan variabel ***y_binary*** yang merupakan transformasi target menjadi dua kelas, kelas **0** untuk **tidak ada indikasi penyakit jantung** dan kelas **1** untuk **ada indikasi penyakit jantung**. Kedua pendekatan ini digunakan untuk **membandingkan kedua model** dan mengetahui **pendekatan yang lebih tepat**.



num	y_binary
0	0
2	1
1	1
0	0
0	0
...	...
1	1
1	1
2	1
3	1
1	1

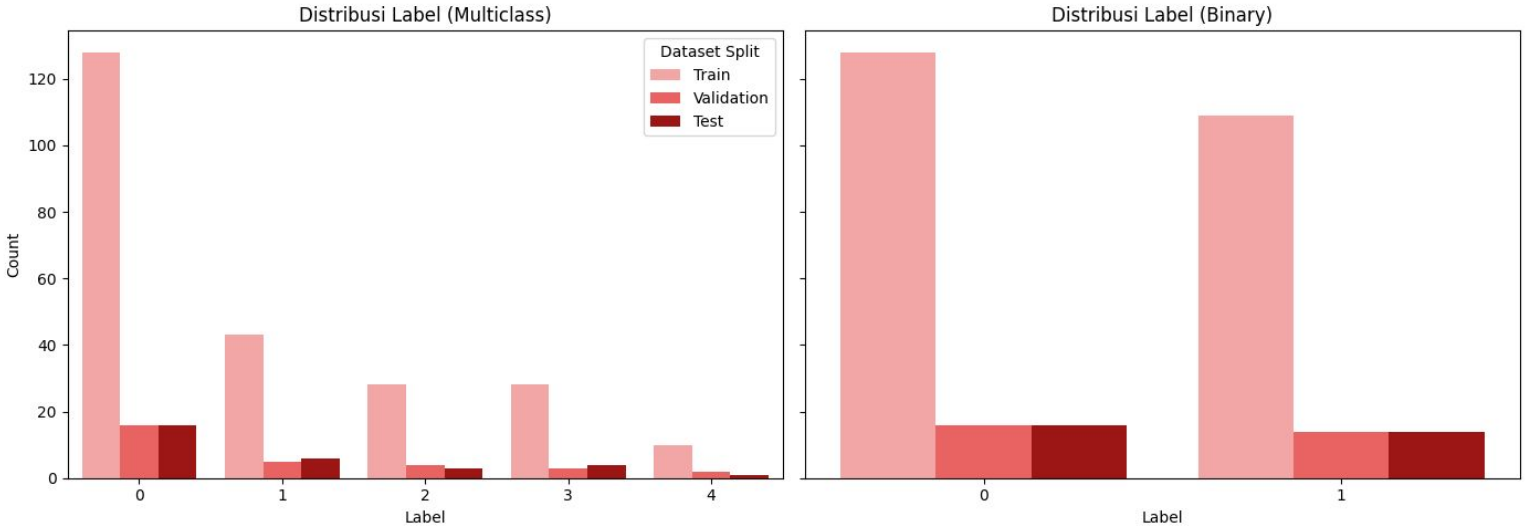
Split Data

Setelah membagi kasus target, kami melakukan ***split data*** untuk kedua kasus

80% data latih (training)

10% data validasi (*validation*)

10% data uji (*test*)



Handling Imbalance Data

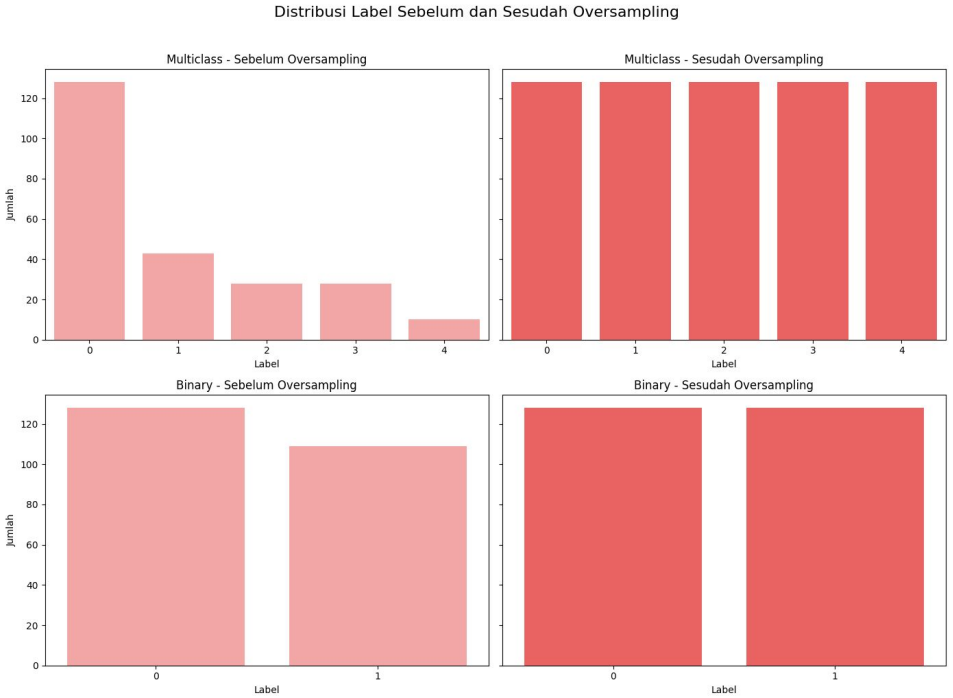
Setelah split data, terlihat bahwa **distribusi label** baik pada kedua kasus masih **belum seimbang**.

Dataset terdiri atas variabel **numerik** dan **kategorik**

Random Oversampler

Pada **data latih (*train*)** untuk menghindari ***data leakage***.

Data leakage (atau kebocoran data) terjadi ketika **data pelatihan** mengandung **informasi** tentang **target** yang sebenarnya tidak akan tersedia saat model digunakan untuk prediksi nyata.



Cara Kerja Random OverSampler

Random OverSampler bertujuan untuk menyeimbangkan distribusi kelas dengan **menyalin ulang data** dari kelas **minoritas secara acak** hingga jumlahnya setara dengan kelas mayoritas.

Langkah-Langkah Random OverSampler

$$N_{\max} = \max_{k \in \{1,\ldots,K\}} N_k$$

$$\Delta_k = N_{\max} - N_k$$

$$\{x_i^{(k)}\}_{i=1}^{\Delta_k} \sim \text{Sample}\,(D_k, \text{size} = \Delta_k, \text{with replacement})$$

$$D' = D \cup \bigcup_{k=1}^{K} \{x_i^{(k)}\}_{i=1}^{\Delta_k}$$

Menghitung Jumlah Data Maksimum tiap Kelas

Hitung **jumlah data maksimum** dalam satu kelas.

Menghitung Jumlah Data Tambahan tiap Kelas

Untuk **setiap kelas** k, hitung **jumlah data** yang perlu ditambahkan. Apabila **selisihnya nol** maka kelas **tidak diubah**.

Sampling Acak dengan Pengembalian

Untuk **setiap kelas** yang memiliki **selisih lebih dari nol**, **ambil sampel** dari data kelas sesuai jumlah **secara acak** dengan pengembalian.

Gabungkan ke Dataset Awal

Gabungkan semua data **hasil sampling** ke **dataset asli.**

Komponen/Parameter pada Model ANN

Layer	
Dense	Layer yang menghubungkan setiap neuron ke semua neuron di layer sebelumnya. Digunakan untuk memproses dan meneruskan informasi dalam jaringan.
Dropout	Teknik regularisasi yang secara acak menonaktifkan sejumlah neuron saat training untuk mencegah overfitting dan membuat model lebih general.
Activation Function	
ReLU (Rectified Linear Unit)	Fungsi aktivasi untuk hidden layer yang menghasilkan 0 jika input negatif, dan input itu sendiri jika positif. Membantu model belajar hubungan non-linear.
Sigmoid	Fungsi aktivasi untuk output layer pada klasifikasi biner. Mengubah output menjadi nilai antara 0 dan 1, seperti probabilitas.
Softmax	Fungsi aktivasi untuk output layer pada klasifikasi multi-kelas. Mengubah output menjadi distribusi probabilitas, sehingga total probabilitas seluruh kelas bernilai 1..
Optimizer	
Adam (Adaptive Moment Estimation)	Algoritma optimisasi populer yang menyesuaikan laju pembelajaran berdasarkan rata-rata gradien dan momentumnya. Cepat dan efisien untuk banyak masalah.
Loss Function	
Binary Crossentropy	Digunakan untuk masalah klasifikasi biner. Mengukur selisih antara label asli dan probabilitas output dari sigmoid. Cocok ketika target hanya dua kelas (0 dan 1).
Sparse Categorical Crossentropy	Digunakan untuk klasifikasi multi-kelas dengan label integer (bukan one-hot encoded). Mengukur selisih antara label asli dan output probabilitas dari softmax.
Metrics	
Accuracy	Mengukur persentase prediksi model yang benar dibandingkan dengan total data. Digunakan untuk evaluasi performa model.

Baseline Model

Multiclass Classification

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(len(y.unique()), activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

Activation Function

Output layer menggunakan ***softmax*** karena ini adalah klasifikasi **multikelas** dan menghasilkan probabilitas untuk setiap kelas.

Jumlah Neuron pada Output Layer

len(y.unique()) menghitung **jumlah kelas target** y

Loss Function

loss='sparse_categorical_crossentropy' digunakan saat label **target** berupa ***integer*** dan cocok untuk klasifikasi **multikelas**.

Binary Classification

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train_binary.shape[1],)),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

model.summary()
```

Activation Function

Output layer menggunakan ***sigmoid*** karena ini adalah klasifikasi **biner** dan menghasilkan probabilitas antara 0 dan 1 untuk satu kelas positif.

Jumlah Neuron pada Output Layer

Satu karena hanya ada **2 kelas** (positif dan negatif).

Loss Function

loss='binary_crossentropy' digunakan saat label **target** berupa ***integer*** **(0 atau 1)** dan cocok untuk klasifikasi **biner**.

Hyperparameter Tuning

Proses **mencari kombinasi terbaik** dari **parameter** yang tidak dipelajari langsung oleh model (seperti **jumlah *neuron*** dan ***dropout***) agar **performa model optimal**. Dalam kasus ini, tuning dilakukan menggunakan ***library* Keras Tuner**, yang memudahkan eksplorasi dan pencarian *hyperparameter* secara otomatis.

Search space:

```
units=hp.Int('units1', min_value=32, max_value=128, step=16)
dropout=hp.Float('dropout1', min_value=0.1, max_value=0.5, step=0.1)

units=hp.Int('units2', min_value=16, max_value=64, step=16)
dropout=hp.Float('dropout2', min_value=0.1, max_value=0.5, step=0.1)
```

RandomSearch

```
tuner = kt.RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=10,
    executions_per_trial=1,
    directory='my_dir',
    project_name='hcv_tuning',
    overwrite=True
)
```

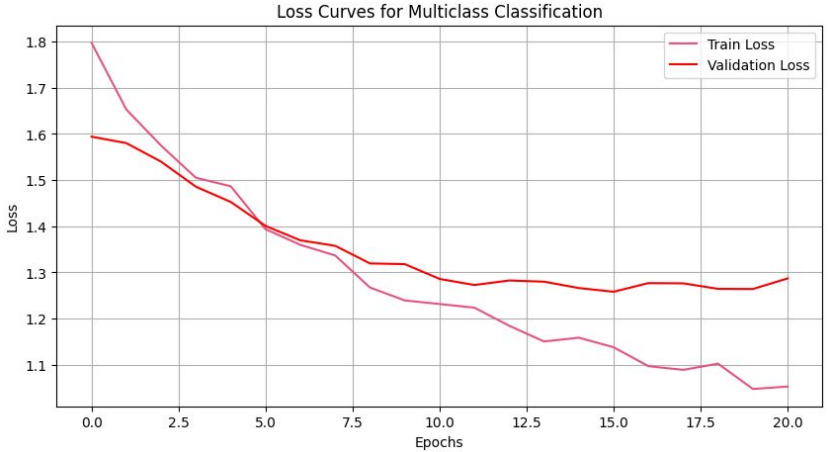
RandomSearch menjalankan tuning dengan **mencoba kombinasi *hyperparameter*** dari ***search space*** secara acak. Didefinisikan **maksimum *trials*** sebanyak **10 kali. *Best Model*** di**evaluasi** berdasarkan ***val_accuracy***. Seluruh **kombinasi parameter disimpan** pada folder *directory* ***my_dir/hcv_tuning***. ***Best parameter*** terpilih adalah parameter dengan ***val_accuracy* tertinggi** dengan ***trial* paling awal**.

Tuning Multi-class classification:

	units1	dropout1	units2	dropout2	val_accuracy	
0	112	0.1	64	0.3	0.600000	
1	32	0.2	32	0.1	0.533333	
2	80	0.1	64	0.2	0.533333	
3	80	0.5	16	0.5	0.600000	
4	112	0.5	32	0.1	0.600000	
5	64	0.3	16	0.4	0.633333	
6	32	0.3	16	0.4	0.600000	
7	80	0.3	32	0.3	0.666667	*Best hyperparameters*
8	48	0.1	16	0.4	0.533333	
9	128	0.4	64	0.3	0.600000	

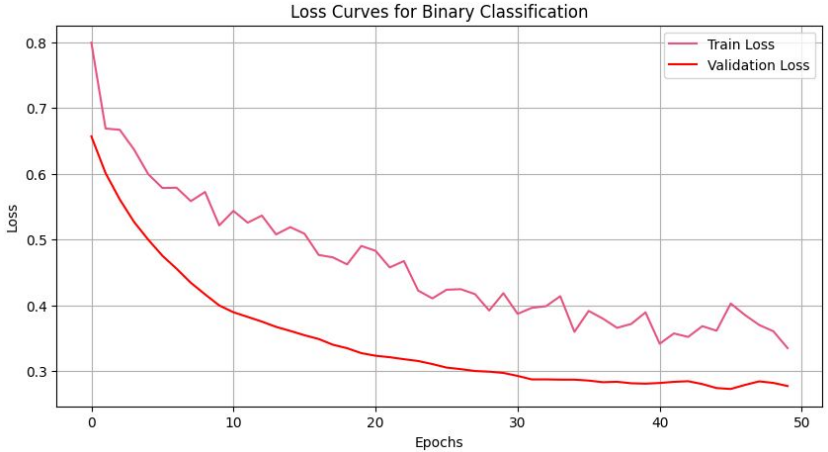
Tuning Binary classification:

	units1	dropout1	units2	dropout2	val_accuracy	
0	64	0.3	32	0.4	0.900000	*Best hyperparameters*
1	64	0.1	16	0.3	0.900000	
2	32	0.2	48	0.2	0.866667	
3	80	0.5	64	0.2	0.900000	
4	32	0.4	32	0.1	0.833333	
5	96	0.2	64	0.1	0.866667	
6	64	0.4	64	0.4	0.900000	
7	128	0.5	48	0.5	0.866667	
8	96	0.3	48	0.4	0.866667	
9	80	0.4	64	0.4	0.900000	



Multiclass classification

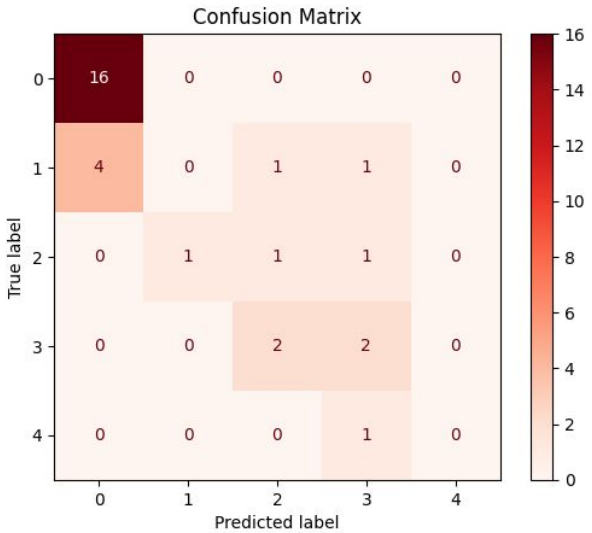
Loss Curve menunjukkan **penurunan** yang **konsisten** pada **data pelatihan dan validasi** di **awal** pelatihan. Setelah sekitar epoch ke-10, ***train loss* terus menurun** sementara ***validation loss* cenderung stabil**. Hal ini menunjukkan bahwa peningkatan performa pada data pelatihan tidak lagi diikuti oleh data validasi.



Binary classification

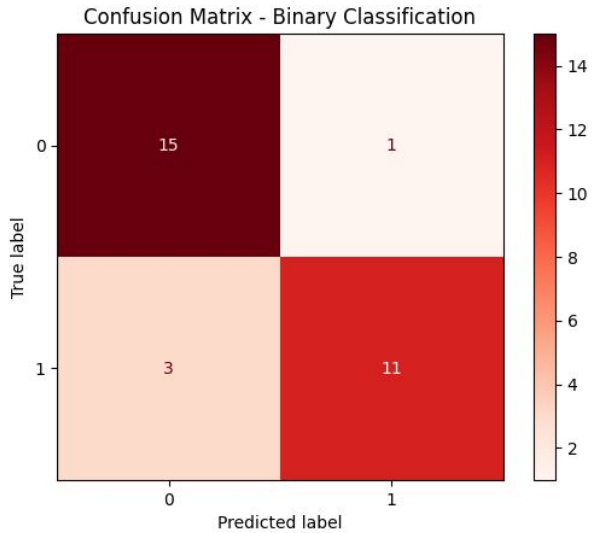
Loss Curve menunjukkan **penurunan** yang **konsisten** pada ***train loss*** dan ***validation loss* selama pelatihan**, terutama di awal. ***Validation loss* terus menurun** secara stabil hingga akhir *epoch*, sementara ***train loss* mengalami fluktuasi ringan**. Pola ini menunjukkan bahwa **model terus membaik** pada **data pelatihan** maupun **validasi sepanjang proses pelatihan**.

Confusion Matrix Multiclass Classification



Dari 30 data uji pasien, model klasifikasi penyakit jantung sangat **akurat mengenali pasien tanpa penyakit** (kelas 0), namun **gagal mengenali kasus** dengan tingkat keparahan lebih tinggi (**kelas 1–4**), yang menunjukkan **bias** terhadap **kelas mayoritas** dan **perlunya perbaikan** seperti penyeimbangan data.

Confusion Matrix Binary Classification



Dari 30 pasien data uji, model mengklasifikasikan **15 pasien sehat dengan benar** (True Negative) dan **11 pasien sakit dengan benar** (True Positive). Namun, ada **3 pasien sakit yang salah diprediksi sehat** (False Negative) dan **1 pasien sehat yang salah diprediksi sakit** (False Positive). Model ini memiliki **tingkat kesalahan kecil**, terutama meminimalkan False Positive, yang penting agar **pasien** sehat **tidak mengalami kekhawatiran yang tidak perlu**.

Class/ Metric	Multiclass Classification					Binary Classification				
	Precission	Sensitivity (Recall)	Specificity	F1-Score	AUC	Precission	Sensitivity (Recall)	Specificity	F1-Score	AUC
0	0.84	1.0000	0.7857	0.91	0.9107	0.83	0.9375	0.7857	0.88	0.9375
1	1.00	0.1667	1.0000	0.29	0.7014	0.92	0.7857	0.9375	0.85	0.9375
2	0.20	0.3333	0.8519	0.25	0.8272	-	-	-	-	-
3	0.00	0.0000	0.9615	0.00	0.8942	-	-	-	-	-
4	0.00	0.0000	0.8621	0.00	0.7931	-	-	-	-	-
Accuracy	0.60	-	-	-	-	0.87	-	-	-	-
Macro Average	0.41	0.30	-	0.29	-	0.88	0.86	-	0.86	-
Weighted Average	0.67	0.60	-	0.57	-	0.87	0.87	-	0.87	-

Insight

Klasifikasi ***multiclass***, model **hanya** berhasil **mengenali** kelas **0 dengan baik** (*precision*, *recall*, F1 > 0.8), sementara kelas lain, terutama kelas 4, tidak terdeteksi sama sekali dan kelas 1, 2, serta 3 memiliki sensitivitas sangat rendah atau nol. **Akurasi keseluruhan** hanya **60%** dengan **macro F1-*score* 0.29**, menunjukkan **ketidakseimbangan** dan **dominasi kelas mayoritas**.

Insight

Klasifikasi ***binary*** menunjukkan **performa jauh lebih baik** dan stabil dengan **akurasi sekitar 87%** serta metrik *precision*, *sensitivity*, *specificity*, F1-*score*, dan AUC tinggi (0.87–0.93). Model ***binary*** **lebih efektif** dan **siap** digunakan untuk **skrining awal penyakit**.

Perbandingan

Multiclass

Target terdiri dari **5 kelas** (0 = tidak ada, 1–4 = tingkat keparahan penyakit jantung)

Akurasi model hanya sekitar **60%**. Model **cenderung bias** ke **kelas mayoritas** (kelas 0).

Kurang cocok untuk aplikasi nyata karena **kinerja** yang **tidak konsisten** dan **sulit diinterpretasikan** akibat **banyaknya kelas** dengan **performa rendah**.

Binary

Target terdiri dari **2 kelas** (0 = tidak ada penyakit, 1 = ada penyakit).

Akurasi model cukup tinggi sekitar **87%**. Model **seimbang** dan dapat mengenali **kedua kelas** dengan baik.

Lebih siap digunakan untuk skrining awal, terutama karena **hasilnya stabil**, **mudah diinterpretasi**, dan metrik evaluasinya **unggul** di semua aspek.

Insight

Penanganan missing values, scaling, dan outlier penting untuk **menjaga kualitas data** dan **meningkatkan performa** model.

RandomOverSampler efektif mengatasi **ketidakseimbangan data** dengan menambah sampel pada kelas minoritas, khususnya pada binary classification.

Penggunaan **Keras Tuner** membantu **menemukan kombinasi** jumlah neuron, dropout, dan hidden layer yang **optimal** dengan **cepat** dan **efisien**.

Kesimpulan

Secara umum, **keberhasilan model klasifikasi** bergantung pada kualitas dan **preprocessing data**, serta **tuning** yang tepat. Pada kasus ini, model **klasifikasi biner** menghasilkan model dengan **performa** yang **baik**. Model ini mampu **mengidentifikasi pola** dengan **akurat** dan hanya sedikit melakukan kesalahan prediksi. Penggunaan model ini dapat **meningkatkan efisiensi** dan kecepatan dalam **proses skrining awal** dan **pengambilan keputusan**.

Tanpa Machine Learning

Data pasien didapat oleh dokter (Usia, tekanan darah, kolesterol, detak jantung, dll)

Analisis dilakukan secara manual oleh dokter (Berdasarkan pengalaman dan observasi gejala)

Hasil penilaian bisa berbeda antar dokter karena variasi pengalaman dan metode observasi.

Risiko salah diagnosis tinggi

Pasien sehat bisa dikira sakit → overdiagnosis

Pasien sakit bisa tidak terdeteksi → underdiagnosis

Penanganan tidak tepat:
- Pasien yang membutuhkan bisa tertunda
- Pasien sehat bisa dirawat tanpa perlu

Dampak akhir

Ketepatan rendah, keputusan tidak konsisten

Biaya tinggi, risiko kesehatan meningkat

Dengan Machine Learning

Dataset pasien dikumpulkan (Format digital: usia, tekanan darah, kolesterol, dll)

Model ANN dilatih menggunakan data historis pasien (Mengenali pola pasien sehat vs sakit dari data sebelumnya)

Model digunakan untuk prediksi otomatis pasien baru (Input: data medis → Output: risiko penyakit jantung)

Dokter menerima hasil prediksi sebagai skrining awal
→ Bisa fokus pada pasien berisiko tinggi
→ Tidak melewatkan pasien dengan gejala samar

Dampak akhir

Penanganan pasien risiko tinggi lebih cepat, meningkatkan peluang hidup.

Efisiensi sumber daya dan penghematan biaya operasional rumah sakit.

Daftar Pustaka

Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository: Heart Disease Dataset*. University of California, School of Information and Computer Science. Retrieved from https://archive.ics.uci.edu/ml/datasets/heart+Disease

Damarla, R. (2022). *Heart Disease Prediction* [Dataset]. Kaggle. Retrieved from https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction

Mackay, J., & Mensah, G. A. (2004). *The atlas of heart disease and stroke*. World Health Organization. Retrieved from https://apps.who.int/iris/handle/10665/43007

Yusuf, S., Hawken, S., Ounpuu, S., Dans, T., Avezum, A., Lanas, F., ... INTERHEART Study Investigators. (2004). Effect of potentially modifiable risk factors associated with myocardial infarction in 52 countries (the INTERHEART study): Case-control study. *The Lancet, 364*(9438), 937–952. https://doi.org/10.1016/S0140-6736(04)17018-9

National Heart, Lung, and Blood Institute. (2023). *What is coronary heart disease?* Retrieved from https://www.nhlbi.nih.gov/health/coronary-heart-disease

American Heart Association. (2021). Heart disease and stroke statistics—2021 update. *Circulation, 143*(8), e254–e743. https://doi.org/10.1161/CIR.0000000000000950

D'Agostino, R. B., Vasan, R. S., Pencina, M. J., Wolf, P. A., Cobain, M., Massaro, J. M., & Kannel, W. B. (2008). General cardiovascular risk profile for use in primary care: The Framingham Heart Study. *Circulation, 117*(6), 743–753. https://doi.org/10.1161/CIRCULATIONAHA.107.699579


THANK YOU!