Nama : Nani Fitria Ramadhani

NIM : 23031554013

Kelas : 2023 D

---Preprocessing Data Mining---

DATASET

```
pip install ucimlrepo
```

```
Requirement already satisfied: ucimlrepo in
/usr/local/lib/python3.11/dist-packages (0.0.7)
Requirement already satisfied: pandas>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in
/usr/local/lib/python3.11/dist-packages (from ucimlrepo) (2025.1.31)
Requirement already satisfied: numpy>=1.23.2 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0-
>ucimlrepo) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0-
>ucimlrepo) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0-
>ucimlrepo) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas>=1.0.0-
>ucimlrepo) (2025.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2-
>pandas>=1.0.0->ucimlrepo) (1.17.0)
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
support2 = fetch_ucirepo(id=880)

# data (as pandas dataframes)
X = support2.data.features
y = support2.data.targets

# metadata
print(support2.metadata)

# variable information
print(support2.variables)
```

{'uci_id': 880, 'name': 'SUPPORT2', 'repository_url': 'https://archive.ics.uci.edu/dataset/880/support2', 'data_url': 'https://archive.ics.uci.edu/static/public/880/data.csv', 'abstract': "This dataset comprises 9105 individual critically ill patients across 5 United States medical centers, accessioned throughout 1989-1991 and 1992-1994.\nEach row concerns hospitalized patient records who met the inclusion and exclusion criteria for nine disease categories: acute respiratory failure, chronic obstructive pulmonary disease, congestive heart failure, liver disease, coma, colon cancer, lung cancer, multiple organ system failure with malignancy, and multiple organ system failure with sepsis. The goal is to determine these patients' 2- and 6-month survival rates based on several physiologic, demographics, and disease severity information. \nIt is an important problem because it addresses the growing national concern over patients' loss of control near the end of life. It enables earlier decisions and planning to reduce the frequency of a mechanical, painful, and prolonged dying process.", 'area': 'Health and Medicine', 'tasks': ['Classification', 'Regression', 'Other'], 'characteristics': ['Tabular', 'Multivariate'], 'num_instances': 9105, 'num_features': 42, 'feature_types': ['Real', 'Categorical', 'Integer'], 'demographics': ['Age', 'Sex', 'Education Level', 'Income', 'Race'], 'target_col': ['death', 'hospdead', 'sfdm2'], 'index_col': ['id'], 'has_missing_values': 'yes', 'missing_values_symbol': 'NaN', 'year_of_dataset_creation': 1995, 'last_updated': 'Mon Sep 09 2024', 'dataset_doi': '10.3886/ICPSR02957.v2', 'creators': ['Frank Harrel'], 'intro_paper': {'ID': 298, 'type': 'NATIVE', 'title': 'A controlled trial to improve care for seriously ill hospitalized patients. The study to understand prognoses and preferences for outcomes and risks of treatments (SUPPORT)', 'authors': 'The SUPPORT Principal Investigators', 'venue': 'In the Journal of the American Medical Association, 274(20):1591–1598', 'year': 1995, 'journal': None, 'DOI': None, 'URL': 'https://pubmed.ncbi.nlm.nih.gov/7474243/', 'sha': None, 'corpus': None, 'arxiv': None, 'mag': None, 'acl': None, 'pmid': None, 'pmcid': None}, 'additional_info': {'summary': "Data sources are medical records, personal interviews, and the National Death Index (NDI). For each patient administrative records data, clinical data and survey data were collected.\nThe objective of the SUPPORT project was to improve decision-making in order to address the growing national concern over the loss of control that patients have near the end of life and to reduce the frequency of a mechanical, painful, and prolonged process of dying. SUPPORT comprised a two-year prospective observational study (Phase I) followed by a two-year controlled clinical trial (Phase II). Phase I of SUPPORT collected data from patients accessioned during 1989-1991 to characterize the care, treatment preferences, and patterns of decision-making among critically ill patients. It also served as a preliminary step for devising an intervention strategy for improving critically-ill patients' care and for the construction of statistical models for predicting patient prognosis and functional status. An intervention was implemented in Phase II of SUPPORT, which accessioned patients

during 1992-1994. The Phase II intervention provided physicians with accurate predictive information on future functional ability, survival probability to six months, and patients' preferences for end-of-life care. Additionally, a skilled nurse was provided as part of the intervention to elicit patient preferences, provide prognoses, enhance understanding, enable palliative care, and facilitate advance planning. The intervention was expected to increase communication, resulting in earlier decisions to have orders against resuscitation, decrease time that patients spent in undesirable states (e.g., in the Intensive Care Unit, on a ventilator, and in a coma), increase physician understanding of patients' preferences for care, decrease patient pain, and decrease hospital resource use. Data collection in both phases of SUPPORT consisted of questionnaires administered to patients, their surrogates, and physicians, plus chart reviews for abstracting clinical, treatment, and decision information. Phase II also collected information regarding the implementation of the intervention, such as patient-specific logs maintained by nurses assigned to patients as part of the intervention. SUPPORT patients were followed for six months after inclusion in the study. Those who did not die within six months or were lost to follow-up were matched against the National Death Index to identify deaths through 1997. Patients who did not die within one year or were lost to follow-up were matched against the National Death Index to identify deaths through 1997.\nAll patients in five United States medical centers who met inclusion and exclusion criteria for nine disease categories: acute respiratory failure, chronic obstructive pulmonary disease, congestive heart failure, liver disease, coma, colon cancer, lung cancer, multiple organ system failure with malignancy, and multiple organ system failure with sepsis. SUPPORT is a combination of patients from 2 studies, each of which lasted 2 years. The first phase concerns 4,301 patients, whereas the second phase concerns 4,804 patients. Time wise, these studies were accessioned in 1989 (June 12) through 1991 (June 11) for phase I and in 1992 (January 7) through 1994 (January 24).", 'purpose': "To develop and validate a prognostic model that estimates survival over a 180-day period for seriously ill hospitalized adults (phase I of SUPPORT) and to compare this model's predictions with those of an existing prognostic system and with physicians' independent estimates (SUPPORT phase II).", 'funded_by': 'Funded by the Robert Wood Johnson Foundation\n', 'instances_represent': 'The instances represent records of critically ill patients admitted to United States hospitals with advanced stages of serious illness.\n', 'recommended_data_splits': 'No recommendation, standard train-test split could be used. Can use three-way holdout split (i.e., train-validation-test) when doing model selection.', 'sensitive_data': 'Yes. There is information about race, gender, income, and education level.', 'preprocessing_description': 'No. Due to the high percentage of missing values, there are a couple of recommended imputation values:\nAccording to the HBiostat Repository (https://hbiostat.org/data/repo/supportdesc, Professor Frank Harrell) the following default values have been found to be useful in imputing

missing baseline physiologic data:\nBaseline Variable\tNormal Fill-in Value\n- Serum albumin (alb)\t3.5\n- PaO2/FiO2 ratio (pafi) \t333.3\n- Bilirubin (bili)\t1.01\n- Creatinine (crea)\t1.01\n- bun\t6.51\n- White blood count (wblc)\t9 (thousands)\n- Urine output (urine)\t2502\nThere are 159 patients surviving 2 months for whom there were no patient or surrogate interviews. These patients have missing sfdm2.', 'variable_info': None, 'citation': 'Please acknowledge the source of this dataset as being from Vanderbilt University Department of Biostatistics, Professor Frank Harrell 2022, url: https://hbiostat.org/data/'}, 'external_url': 'https://hbiostat.org/data'}

	name	role	type	demographic	\
0	id	ID	Integer	None	
1	age	Feature	Continuous	Age	
2	death	Target	Continuous	None	
3	sex	Feature	Categorical	Sex	
4	hospdead	Target	Binary	None	
5	slos	Other	Continuous	None	
6	d.time	Other	Continuous	None	
7	dzgroup	Feature	Categorical	None	
8	dzclass	Feature	Categorical	None	
9	num.co	Feature	Continuous	None	
10	edu	Feature	Categorical	Education Level	
11	income	Feature	Categorical	Income	
12	scoma	Feature	Continuous	None	
13	charges	Feature	Continuous	None	
14	totcst	Feature	Continuous	None	
15	totmcst	Feature	Continuous	None	
16	avtisst	Feature	Continuous	None	
17	race	Feature	Categorical	Race	
18	sps	Feature	Continuous	None	
19	aps	Feature	Continuous	None	
20	surv2m	Feature	Continuous	None	
21	surv6m	Feature	Continuous	None	
22	hday	Feature	Integer	None	
23	diabetes	Feature	Continuous	None	
24	dementia	Feature	Continuous	None	
25	ca	Feature	Categorical	None	
26	prg2m	Feature	Continuous	None	
27	prg6m	Feature	Categorical	None	
28	dnr	Feature	Categorical	None	
29	dnrday	Feature	Continuous	None	
30	meanbp	Feature	Continuous	None	
31	wblc	Feature	Continuous	None	
32	hrt	Feature	Continuous	None	
33	resp	Feature	Continuous	None	
34	temp	Feature	Continuous	None	
35	pafi	Feature	Continuous	None	
36	alb	Feature	Continuous	None	
37	bili	Feature	Continuous	None	

```
38      crea  Feature   Continuous             None
39       sod  Feature   Continuous             None
40        ph  Feature   Continuous             None
41   glucose  Feature      Integer             None
42       bun  Feature      Integer             None
43     urine  Feature      Integer             None
44      adlp  Feature  Categorical             None
45      adls  Feature   Continuous             None
46     sfdm2   Target  Categorical             None
47     adlsc  Feature   Continuous             None

                                              description  units
missing_values
0                                                    None   None
no
1                          Age of the patients in years  years
no
2   Death at any time up to National Death Index (...   None
no
3   Gender of the patient. Listed values are {male...   None
no
4                                     Death in hospital   None
no
5                   Days from Study Entry to Discharge   None
no
6                                     Days of follow-up   None
no
7   The patient's disease sub category amogst ARF/...   None
no
8   The patient's disease category amongst "ARF/MO...   None
no
9   The number of simultaneous diseases (or comorb...   None
no
10                                   Years of education  years
yes
11  Income of the patient. Listed values are {"$11...   None
yes
12  SUPPORT day 3 Coma Score based on Glasgow scal...   None
yes
13                                     Hospital charges   None
yes
14         Total ratio of costs to charges (RCC) cost   None
yes
15                                     Total micro cost   None
yes
16  Average TISS score, days 3-25, where Therapeut...   None
yes
17  Race of the patient. Listed values are {asian,...   None
yes
18  SUPPORT physiology score on day 3 (predicted b...   None
```

```
yes
19  APACHE III day 3 physiology score (no coma, im...   None
yes
20  SUPPORT model 2-month survival estimate at day...   None
yes
21  SUPPORT model 6-month survival estimate at day...   None
yes
22    Day in hospital at which patient entered study.   None
no
23  Whether the patient exhibits diabetes (Com 27-...   None
no
24  Whether the patient exhibits dementia (Comorbi...   None
no
25  Whether the patient has cancer (yes), whether ...   None
no
26  Physician’s 2-month survival estimate for pati...   None
yes
27  Physician’s 6-month survival estimate for pati...   None
yes
28  Whether the patient has a do not rescuscitate ...   None
yes
29              Day of DNR order (<0 if before study)   None
yes
30  mean arterial blood pressure of the patient, m...   None
yes
31  counts of white blood cells (in thousands) mea...   None
yes
32       heart rate of the patient measured at day 3.   None
yes
33  respiration rate of the patient measured at da...   None
yes
34  temperature in Celsius degrees measured at day 3.   None
no
35  $PaO_2/FiO_2$ ratio measured at day 3. The rat...   None
yes
36            serum albumin levels measured at day 3.   None
yes
37                bilirubin levels measured at day 3.   None
yes
38         serum creatinine levels measured at day 3.   None
yes
39      serum sodium concentration measured at day 3.   None
yes
40  Arterial blood pH. The pH of blood is usually ...   None
yes
41                   Glucose levels measured at day 3.   None
yes
42      Blood urea nitrogen levels measured at day 3.   None
yes
43                     Urine output measured at day 3.   None
```

```
yes
44  Index of Activities of Daily Living (ADL) of t...   None
yes
45  Index of Activities of Daily Living (ADL) of t...   None
yes
46  Level of functional disability of the patient ...   None
yes
47                Imputed ADL Calibrated to Surrogate.   None
no
```

```
import pandas as pd
df =
pd.read_csv('https://archive.ics.uci.edu/static/public/880/data.csv')
df.head()
```

```
{"type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9105 entries, 0 to 9104
Data columns (total 48 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        9105 non-null   int64
 1   age       9105 non-null   float64
 2   death     9105 non-null   int64
 3   sex       9105 non-null   object
 4   hospdead  9105 non-null   int64
 5   slos      9105 non-null   int64
 6   d.time    9105 non-null   int64
 7   dzgroup   9105 non-null   object
 8   dzclass   9105 non-null   object
 9   num.co    9105 non-null   int64
 10  edu       7471 non-null   float64
 11  income    6123 non-null   object
 12  scoma     9104 non-null   float64
 13  charges   8933 non-null   float64
 14  totcst    8217 non-null   float64
 15  totmcst   5630 non-null   float64
 16  avtisst   9023 non-null   float64
 17  race      9063 non-null   object
 18  sps       9104 non-null   float64
 19  aps       9104 non-null   float64
 20  surv2m    9104 non-null   float64
 21  surv6m    9104 non-null   float64
 22  hday      9105 non-null   int64
 23  diabetes  9105 non-null   int64
 24  dementia  9105 non-null   int64
 25  ca        9105 non-null   object
```

```
 26  prg2m     7456 non-null   float64
 27  prg6m     7472 non-null   float64
 28  dnr       9075 non-null   object
 29  dnrday    9075 non-null   float64
 30  meanbp    9104 non-null   float64
 31  wblc      8893 non-null   float64
 32  hrt       9104 non-null   float64
 33  resp      9104 non-null   float64
 34  temp      9104 non-null   float64
 35  pafi      6780 non-null   float64
 36  alb       5733 non-null   float64
 37  bili      6504 non-null   float64
 38  crea      9038 non-null   float64
 39  sod       9104 non-null   float64
 40  ph        6821 non-null   float64
 41  glucose   4605 non-null   float64
 42  bun       4753 non-null   float64
 43  urine     4243 non-null   float64
 44  adlp      3464 non-null   float64
 45  adls      6238 non-null   float64
 46  sfdm2     7705 non-null   object
 47  adlsc     9105 non-null   float64
dtypes: float64(31), int64(9), object(8)
memory usage: 3.3+ MB

df.shape

(9105, 48)

import pandas as pd

file_url = 'https://archive.ics.uci.edu/static/public/880/data.csv'
df = pd.read_csv(file_url)

# Cek kolom numerik dan kategorikal
numerical_cols = df.select_dtypes(include=['int64',
'float64']).columns.tolist()
categorical_cols =
df.select_dtypes(include=['object']).columns.tolist()

# Tampilkan hasil
print("Kolom Numerik:", numerical_cols)
print("Kolom Kategorikal:", categorical_cols)

Kolom Numerik: ['id', 'age', 'death', 'hospdead', 'slos', 'd.time',
'num.co', 'edu', 'scoma', 'charges', 'totcst', 'totmcst', 'avtisst',
'sps', 'aps', 'surv2m', 'surv6m', 'hday', 'diabetes', 'dementia',
'prg2m', 'prg6m', 'dnrday', 'meanbp', 'wblc', 'hrt', 'resp', 'temp',
'pafi', 'alb', 'bili', 'crea', 'sod', 'ph', 'glucose', 'bun', 'urine',
'adlp', 'adls', 'adlsc']
```

```
Kolom Kategorikal: ['sex', 'dzgroup', 'dzclass', 'income', 'race',
'ca', 'dnr', 'sfdm2']
```

DATA CLEANING

Cek Duplikat Data. Untuk mngetahui adanya data duplikat

```
df.duplicated().sum()
```

```
0
```

```
output_file = "data_preprocessing.csv"
df.to_csv(output_file, index=False)

print(df.sex.unique())
print(df.dzgroup.unique())
print(df.dzclass.unique())
print(df.income.unique())
print(df.race.unique())
print(df.ca.unique())
print(df.dnr.unique())
print(df.sfdm2.unique())
```

```
['male' 'female']
['Lung Cancer' 'Cirrhosis' 'ARF/MOSF w/Sepsis' 'Coma' 'CHF' 'Colon
Cancer'
 'COPD' 'MOSF w/Malig']
['Cancer' 'COPD/CHF/Cirrhosis' 'ARF/MOSF' 'Coma']
['$11-$25k' 'under $11k' nan '$25-$50k' '>$50k']
['other' 'white' 'black' 'hispanic' 'asian' nan]
['metastatic' 'no' 'yes']
['no dnr' nan 'dnr after sadm' 'dnr before sadm']
[nan '<2 mo. follow-up' 'no(M2 and SIP pres)' 'SIP>=30'
 'adl>=4 (>=5 if sur)' 'Coma or Intub']
```

Cek Missing Value

Untuk menghapus kolom jika terdapat (>50%) data yang hilang dari jumlah total. Mengisi missing value dengan median untuk kolom numerik. Mengisi missing value dengan modus untuk kolom kategorikal.

```
import pandas as pd

file_path = 'data_preprocessing.csv'
df = pd.read_csv(file_path)

# 1. Menghapus kolom dengan lebih dari 50% missing values
threshold = 0.5 * len(df)  # 50% dari jumlah total data
```

```
df = df.dropna(thresh=threshold, axis=1)

# 2. Mengisi missing value untuk kolom numerik dengan median
num_cols = df.select_dtypes(include=['float64', 'int64']).columns
df[num_cols] = df[num_cols].fillna(df[num_cols].median())

# 3. Mengisi missing value untuk kolom kategori dengan modus
cat_cols = df.select_dtypes(include=['object']).columns
for col in cat_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

df.isnull().sum()
```

```
id          0
age         0
death       0
sex         0
hospdead    0
slos        0
d.time      0
dzgroup     0
dzclass     0
num.co      0
edu         0
income      0
scoma       0
charges     0
totcst      0
totmcst     0
avtisst     0
race        0
sps         0
aps         0
surv2m      0
surv6m      0
hday        0
diabetes    0
dementia    0
ca          0
prg2m       0
prg6m       0
dnr         0
dnrday      0
meanbp      0
wblc        0
hrt         0
resp        0
temp        0
pafi        0
alb         0
```

```
bili        0
crea        0
sod         0
ph          0
glucose     0
bun         0
adls        0
sfdm2       0
adlsc       0
dtype: int64
```

```
new_file_path = "data_missing_value.csv"
df.to_csv(new_file_path, index=False)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9105 entries, 0 to 9104
Data columns (total 46 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        9105 non-null   int64
 1   age       9105 non-null   float64
 2   death     9105 non-null   int64
 3   sex       9105 non-null   object
 4   hospdead  9105 non-null   int64
 5   slos      9105 non-null   int64
 6   d.time    9105 non-null   int64
 7   dzgroup   9105 non-null   object
 8   dzclass   9105 non-null   object
 9   num.co    9105 non-null   int64
 10  edu       9105 non-null   float64
 11  income    9105 non-null   object
 12  scoma     9105 non-null   float64
 13  charges   9105 non-null   float64
 14  totcst    9105 non-null   float64
 15  totmcst   9105 non-null   float64
 16  avtisst   9105 non-null   float64
 17  race      9105 non-null   object
 18  sps       9105 non-null   float64
 19  aps       9105 non-null   float64
 20  surv2m    9105 non-null   float64
 21  surv6m    9105 non-null   float64
 22  hday      9105 non-null   int64
 23  diabetes  9105 non-null   int64
 24  dementia  9105 non-null   int64
 25  ca        9105 non-null   object
 26  prg2m     9105 non-null   float64
 27  prg6m     9105 non-null   float64
 28  dnr       9105 non-null   object
```

```
 29  dnrday    9105 non-null   float64
 30  meanbp    9105 non-null   float64
 31  wblc      9105 non-null   float64
 32  hrt       9105 non-null   float64
 33  resp      9105 non-null   float64
 34  temp      9105 non-null   float64
 35  pafi      9105 non-null   float64
 36  alb       9105 non-null   float64
 37  bili      9105 non-null   float64
 38  crea      9105 non-null   float64
 39  sod       9105 non-null   float64
 40  ph        9105 non-null   float64
 41  glucose   9105 non-null   float64
 42  bun       9105 non-null   float64
 43  adls      9105 non-null   float64
 44  sfdm2     9105 non-null   object
 45  adlsc     9105 non-null   float64
dtypes: float64(29), int64(9), object(8)
memory usage: 3.2+ MB
```

Cek Handling Noisy

```
import pandas as pd
import numpy as np

file_path = "data_missing_value.csv"
df = pd.read_csv(file_path)
```

Handling Noisy Data Numerik

Model klasifikasi yang saya gunakan adalah Random Forest. Karena Random Forest mampu menangani hubungan yang lebih kompleks yang hanya mengasumsikan hubungan linier antar variabel. Random Forest juga lebih fleksibel untuk berbagai jenis data.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

file_path = "data_missing_value.csv"
df = pd.read_csv(file_path)

# Membulatkan umur
df["age"] = df["age"].round()

# Fitur yang digunakan untuk prediksi
predictors = ['age', 'scoma', 'temp']
predictors = [col for col in predictors if col in df.columns]

if 'death' in df.columns and len(predictors) > 0:
    # Ambil hanya baris yang memiliki nilai death (tidak NaN)
```

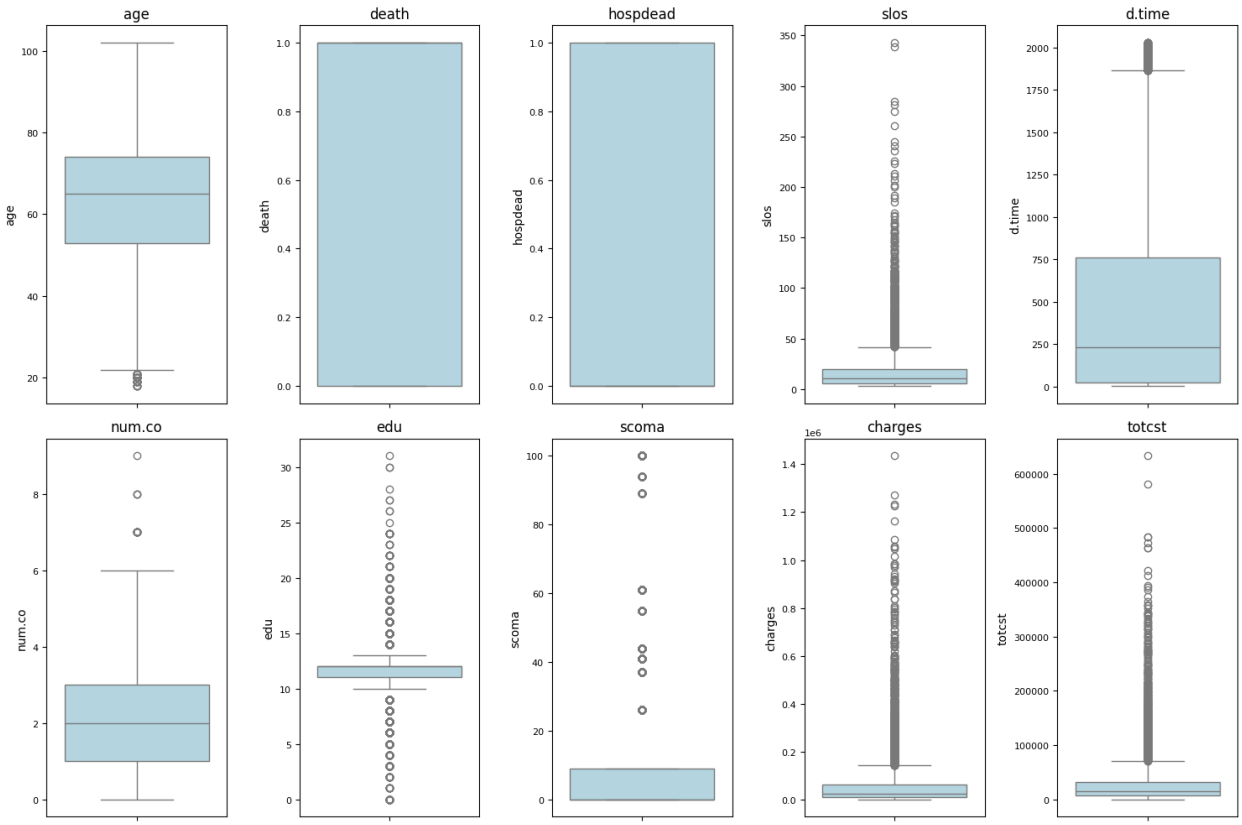
```
    df_clean = df.dropna(subset=['death'] + predictors)

    # Pastikan ada cukup data untuk melatih model
    if not df_clean.empty:
        model = RandomForestClassifier(n_estimators=100,
random_state=42)

        X = df_clean[predictors]
        y = df_clean['death']

        # Latih model
        model.fit(X, y)

        # Ambil baris yang nilai 'death'-nya kosong
        missing_death = df[df['death'].isna()]

        # Pastikan ada data untuk diprediksi
        if not missing_death.empty:
            df.loc[df['death'].isna(), 'death'] =
model.predict(missing_death[predictors])
```

Handling Noisy Data Ketegorikal

Berikut untuk menangani data kategorikal. Mengubah ke huruf kecil dan menghapus spasi berlebih untuk mencegah inkonsistensi data kategori akibat perbedaan kapitalisasi dan spasi. Menghapus nilai yang tidak bermakna (contoh: "unknown" pada kolom race tidak berguna) sehingga mempermudah pengisian data missing values dengan cara yang lebih baik. Menghindari data kosong dengan mengganti NaN menggunakan nilai mode dalam kategorikal.

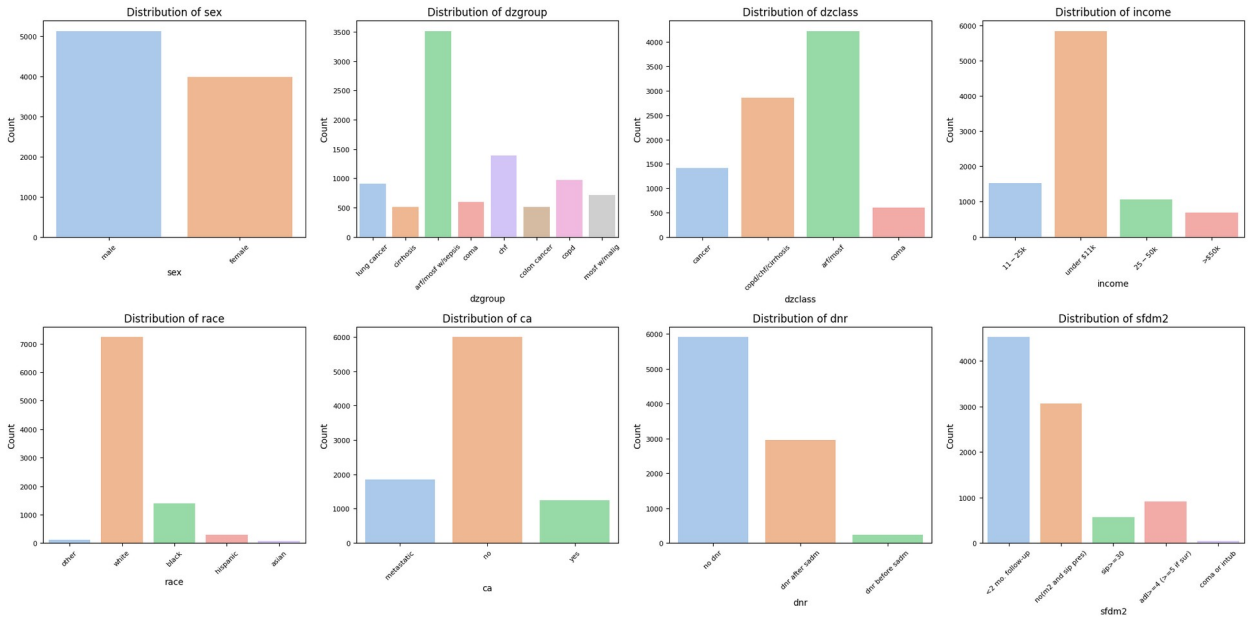
```
categorical_cols = ['sex', 'dzgroup', 'dzclass', 'income', 'race',
'ca', 'dnr', 'sfdm2']

# Lowercase + Hapus Spasi Berlebih
for col in categorical_cols:
    df[col] = df[col].astype(str).str.lower().str.strip()

# Mengatasi Data Tidak Valid ("???", "unknown", "-", dll.)**
invalid_values = ['???', 'unknown', 'none', '-', 'nan', 'null']
for col in categorical_cols:
    df[col] = df[col].apply(lambda x: np.nan if x in invalid_values
else x)

# Mengisi NaN dengan Modus
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])

cleaned_noisy_file = "data_cleaning.csv"
df.to_csv(cleaned_noisy_file, index=False)
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams

file_path = "data_cleaning.csv"
df = pd.read_csv(file_path)

new_df = df.copy()

# Variabel numerik yang akan divisualisasikan
var_num = ['age', 'death', 'hospdead', 'slos', 'd.time', 'num.co',
'edu', 'scoma', 'charges', 'totcst']

# Set ukuran figure
rcParams['figure.figsize'] = 12, 6
rcParams['lines.linewidth'] = 2
rcParams['xtick.labelsize'] = 8
rcParams['ytick.labelsize'] = 8

# Buat subplot grid
num_cols = 5
num_rows = -(-len(var_num) // num_cols)

fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 5 *
num_rows))

# Flatten axes agar mudah diiterasi
axes = axes.flatten()

for i, col in enumerate(var_num):
    if col in new_df.columns:
        sns.boxplot(y=new_df[col], color='lightblue', ax=axes[i])
        axes[i].set_title(col)
    else:
        axes[i].axis("off")

plt.tight_layout()
plt.show()
```




```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

file_path = "data_cleaning.csv"
df = pd.read_csv(file_path)

new_df = df.copy()

# Variabel kategori yang akan divisualisasikan
var_cat = ['sex', 'dzgroup', 'dzclass', 'income', 'race', 'ca', 'dnr',
'sfdm2']

# Atur ukuran gambar lebih besar
fig, axes = plt.subplots(2, 4, figsize=(20, 10))
axes = axes.flatten()

for i, col in enumerate(var_cat):
    if col in new_df.columns:
        sns.countplot(data=new_df, x=col, hue=col, palette="pastel",
legend=False, ax=axes[i])
        axes[i].set_title(f'Distribution of {col}')
        axes[i].set_xlabel(col)
        axes[i].set_ylabel('Count')
        axes[i].tick_params(axis='x', rotation=45)
```

```
    else:
        axes[i].axis("off")

plt.tight_layout()
plt.show()
```




```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9105 entries, 0 to 9104
Data columns (total 46 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   id        9105 non-null   int64
 1   age       9105 non-null   float64
 2   death     9105 non-null   int64
 3   sex       9105 non-null   object
 4   hospdead  9105 non-null   int64
 5   slos      9105 non-null   int64
 6   d.time    9105 non-null   int64
 7   dzgroup   9105 non-null   object
 8   dzclass   9105 non-null   object
 9   num.co    9105 non-null   int64
 10  edu       9105 non-null   float64
 11  income    9105 non-null   object
 12  scoma     9105 non-null   float64
 13  charges   9105 non-null   float64
 14  totcst    9105 non-null   float64
 15  totmcst   9105 non-null   float64
 16  avtisst   9105 non-null   float64
```

```
 17  race      9105 non-null   object
 18  sps       9105 non-null   float64
 19  aps       9105 non-null   float64
 20  surv2m    9105 non-null   float64
 21  surv6m    9105 non-null   float64
 22  hday      9105 non-null   int64
 23  diabetes  9105 non-null   int64
 24  dementia  9105 non-null   int64
 25  ca        9105 non-null   object
 26  prg2m     9105 non-null   float64
 27  prg6m     9105 non-null   float64
 28  dnr       9105 non-null   object
 29  dnrday    9105 non-null   float64
 30  meanbp    9105 non-null   float64
 31  wblc      9105 non-null   float64
 32  hrt       9105 non-null   float64
 33  resp      9105 non-null   float64
 34  temp      9105 non-null   float64
 35  pafi      9105 non-null   float64
 36  alb       9105 non-null   float64
 37  bili      9105 non-null   float64
 38  crea      9105 non-null   float64
 39  sod       9105 non-null   float64
 40  ph        9105 non-null   float64
 41  glucose   9105 non-null   float64
 42  bun       9105 non-null   float64
 43  adls      9105 non-null   float64
 44  sfdm2     9105 non-null   object
 45  adlsc     9105 non-null   float64
dtypes: float64(29), int64(9), object(8)
memory usage: 3.2+ MB
```

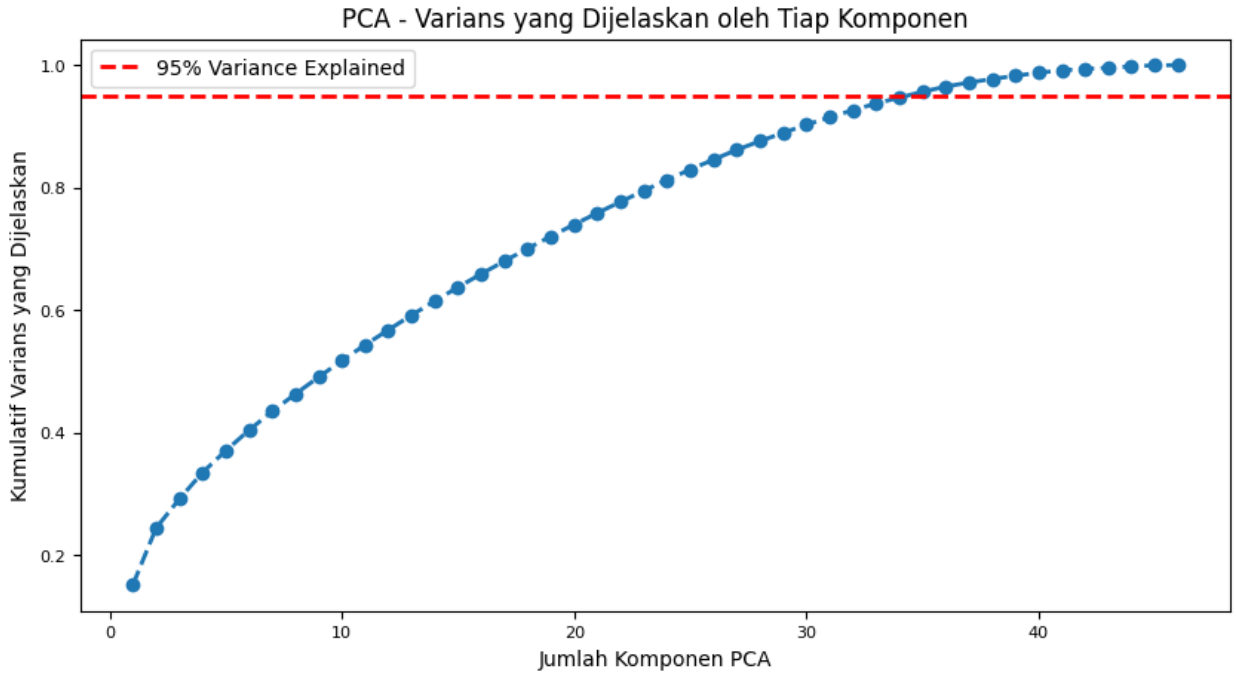
DATA REDUCTION

Reduksi Dimensi dengan PCA (Principal Component Analysis). Dimana tipe data object menggunakan LabelEncoder untuk mengonversi nilai kategorikal menjadi numerik. Kemudian untuk menormalkan data agar memiliki mean = 0 dan standar deviasi = 1 maka menggunakan StandardScaler, hal ini penting karena PCA sensitif terhadap skala data. Menerapkan PCA ke seluruh komponen untuk melihat distribusi varians. Membuat plot scree untuk melihat seberapa banyak varians yang dapat dijelaskan oleh jumlah komponen PCA tertentu. Garis horizontal merah menunjukkan batas 95% varians.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```
file_path = "data_cleaning.csv"
df = pd.read_csv(file_path)

# Encode fitur kategorikal
df_encoded = df.copy()
categorical_cols =
df_encoded.select_dtypes(include=["object"]).columns

for col in categorical_cols:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col])

# Standarisasi data numerik
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_encoded)

# PCA
pca = PCA(n_components=None)  # None = Ambil semua komponen untuk
analisis awal
pca_data = pca.fit_transform(scaled_data)

# Hitung varians yang dijelaskan oleh tiap komponen
explained_variance = np.cumsum(pca.explained_variance_ratio_)

# Plot varians kumulatif
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(explained_variance) + 1), explained_variance,
marker="o", linestyle="--")
plt.xlabel("Jumlah Komponen PCA")
plt.ylabel("Kumulatif Varians yang Dijelaskan")
plt.title("PCA - Varians yang Dijelaskan oleh Tiap Komponen")
plt.axhline(y=0.95, color="r", linestyle="--", label="95% Variance
Explained")
plt.legend()
plt.show()

# Pilih jumlah komponen yang mempertahankan 95% informasi
optimal_components = np.argmax(explained_variance >= 0.95) + 1  #
Ambil jumlah komponen pertama yang mencapai 95%
pca_optimal = PCA(n_components=optimal_components)
pca_result = pca_optimal.fit_transform(scaled_data)

# Dapatkan nama kolom asli
original_columns = df_encoded.columns

# Buat nama baru berdasarkan kontribusi fitur asli
loading_matrix = pca_optimal.components_.T  # Transposisi supaya tiap
baris sesuai fitur asli
important_features =
```

```
[original_columns[np.argmax(np.abs(loading_matrix[:, i]))] for i in
range(optimal_components)]
pca_column_names = [f"PCA_{feat}" for feat in important_features]

pca_df = pd.DataFrame(pca_result, columns=pca_column_names)
pca_df.to_csv("data_pca.csv", index=False)

print(f"menggunakan {optimal_components} komponen yang mempertahankan
95% informasi.")
```




```
menggunakan 35 komponen yang mempertahankan 95% informasi.
```

DATA TRANSFORMATION

Data Transformation adalah proses mengubah format, struktur, atau nilai dari data agar lebih mudah dianalisis sesuai dengan kebutuhan. Karena data sudah direduksi menggunakan model PCA, saya memilih teknik transformasi berupa Normalization yaitu Standardization dan Minmax Normalization. Tetapi saya lebih merekomendasikan menggunakan Standardization karena model PCA sensitif dan lebih stabil terhadap perubahan distribusi data.

```
# Menggunakan Z-score Standardization
import pandas as pd
from sklearn.preprocessing import StandardScaler

file_path = "data_pca.csv"
df = pd.read_csv(file_path)
```

```
standard_scaler = StandardScaler()

df_standard = pd.DataFrame(standard_scaler.fit_transform(df),
columns=df.columns)

df_standard.to_csv("data_pca_standard.csv", index=False)

# Menggunakan Min-Max Normalization
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

file_path = "data_pca.csv"
df = pd.read_csv(file_path)

min_max_scaler = MinMaxScaler()

df_minmax = pd.DataFrame(min_max_scaler.fit_transform(df),
columns=df.columns)

df_minmax.to_csv("data_pca_minmax.csv", index=False)
```