

UAS Pembelajaran Mesin dengan Python

Kelompok :

- 1. Claresta Tirta S. G (6161901025)
- 2. Christian Jauhari (6161901028)
- 3. Fransiska Nadya C. P (6161901051)
- 4. Tiara Alamanda (6161901116)

Preprocessing Data

Load Data

```
In [ ]:

import pandas as pd
url = 'https://raw.githubusercontent.com/blackhespy/heart-disease-data/main/Heart%20Disease.csv'
df = pd.read_csv(url)
df
```

Out[]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	d
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	
...
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	
4238	1	40	3.0	0	0.0	0.0	0	1	0	185.0	141.0	
4239	0	39	3.0	1	30.0	0.0	0	0	0	196.0	133.0	

4240 rows x 16 columns



Sebelumnya sudah di load data csv mengenai sejumlah data terkait penyebab penyakit jantung dan sudah dimasukkan ke dalam variable df. Data ini berjumlah 4240 baris dan 16 kolom.

Langkah-langkah Persiapan Data

```
In [ ]:

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   male                4240 non-null  int64
 1   age                 4240 non-null  int64
```

```

2   education      4135 non-null    float64
3   currentSmoker  4240 non-null    int64
4   cigsPerDay     4211 non-null    float64
5   BPMeds         4187 non-null    float64
6   prevalentStroke 4240 non-null    int64
7   prevalentHyp   4240 non-null    int64
8   diabetes       4240 non-null    int64
9   totChol        4190 non-null    float64
10  sysBP          4240 non-null    float64
11  diaBP          4240 non-null    float64
12  BMI            4221 non-null    float64
13  heartRate      4239 non-null    float64
14  glucose        3852 non-null    float64
15  TenYearCHD     4240 non-null    int64

```

```
dtypes: float64(9), int64(7)
```

```
memory usage: 530.1 KB
```

Lalu dengan menggunakan `df.info()`, dapat dilihat untuk kolom seperti Education, cigsPerDay, BPMeds, totChol, BMI, heartRate, dan glucose terdapat data yang bernilai Null (data kosong) karena jumlah baris yang ditunjukkan tidak sama dengan jumlah baris pada data awal, yaitu 4240 baris. Dapat diperhatikan pula tipe data dari masing-masing kolom.

```
In [ ]:
```

```
df.describe()
```

```
Out[ ]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	di
count	4240.000000	4240.000000	4135.000000	4240.000000	4211.000000	4187.000000	4240.000000	4240.000000	4240.
mean	0.429245	49.580189	1.979444	0.494104	9.005937	0.029615	0.005896	0.310613	0.
std	0.495027	8.572942	1.019791	0.500024	11.922462	0.169544	0.076569	0.462799	0.
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.

```
In [ ]:
```

```
df = df.drop(columns=['education', 'cigsPerDay', 'BPMeds', 'diabetes', 'sysBP', 'diaBP', 'heartRate'])
df
```

```
Out[ ]:
```

	male	age	currentSmoker	prevalentStroke	prevalentHyp	totChol	BMI	glucose	TenYearCHD
0	1	39	0	0	0	195.0	26.97	77.0	0
1	0	46	0	0	0	250.0	28.73	76.0	0
2	1	48	1	0	0	245.0	25.34	70.0	0
3	0	61	1	0	1	225.0	28.58	103.0	1
4	0	46	1	0	0	285.0	23.10	85.0	0
...
4235	0	48	1	0	0	248.0	22.00	86.0	0
4236	0	44	1	0	0	210.0	19.16	NaN	0
4237	0	52	0	0	0	269.0	21.47	107.0	0
4238	1	40	0	0	1	185.0	25.60	72.0	0

4239 male age currentSmoker prevalentStroke prevalentHyp totChol BMI glucose TenYearCHD

4240 rows x 9 columns

Kelompok kami memutuskan untuk menghapus kolom education, cigsPerDay, BPMeds, diabetes, sysBP, diaBP, dan heartRate. Kolom education dihapus karena menurut kami, pendidikan tidak ada hubungannya dengan penyakit jantung. Lalu, kami menghapus kolom cigsPerDay, BPMeds, dan diabetes karena bisa diketahui dari kolom yang lain. Kemudian kolom sysBP, diaBP, dan heartRate dapat dijelaskan oleh kolom prevalentHyp.

Cek Nilai NA pada tiap baris

In []:

```
persentase_data_kosong= df.isna().sum()*100/len(df)
nilaikosong_df= pd.DataFrame({'Persentase Data Kosong': persentase_data_kosong})
nilaikosong_df
```

Out[]:

Persentase Data Kosong	
male	0.000000
age	0.000000
currentSmoker	0.000000
prevalentStroke	0.000000
prevalentHyp	0.000000
totChol	1.179245
BMI	0.448113
glucose	9.150943
TenYearCHD	0.000000

In []:

```
df=df.dropna()
df
```

Out[]:

	male	age	currentSmoker	prevalentStroke	prevalentHyp	totChol	BMI	glucose	TenYearCHD
0	1	39	0	0	0	195.0	26.97	77.0	0
1	0	46	0	0	0	250.0	28.73	76.0	0
2	1	48	1	0	0	245.0	25.34	70.0	0
3	0	61	1	0	1	225.0	28.58	103.0	1
4	0	46	1	0	0	285.0	23.10	85.0	0
...
4234	1	51	1	0	0	207.0	19.71	68.0	0
4235	0	48	1	0	0	248.0	22.00	86.0	0
4237	0	52	0	0	0	269.0	21.47	107.0	0
4238	1	40	0	0	1	185.0	25.60	72.0	0
4239	0	39	1	0	0	196.0	20.91	80.0	0

3828 rows x 9 columns

Pada bagian ini, kami memutuskan untuk menghapus baris yang terdapat nilai Null. Sehingga diperoleh data yang baru dengan jumlah baris adalah 3828 dan jumlah kolom adalah 9.

Logistic Regression

Menentukan data X (response) dan y (class)

In []:

```
x = df.drop(columns='TenYearCHD')
y = df[['TenYearCHD']]

#dfy = np.ravel(dfy)
```

Splitting Data

In []:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0
)
```

Logistic Regression

In []:

```
from sklearn.linear_model import LogisticRegression
logisticRegr = LogisticRegression()
logreg = logisticRegr.fit(x_train, y_train)
logreg
```

```

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max_iter) or scale the data as shown in:
<https://scikit-learn.org/stable/modules/preprocessing.html>
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
extra warning msg= LOGISTIC SOLVER CONVERGENCE MSG,

Out[]:

```
LogisticRegression()
```

In []:

```
print(logreg.coef , logreg.intercept )
```

```
[[0.51352497 0.06106617 0.57604212 0.73701651 0.47277181 0.00107754
  0.01138974 0.00686463]] [-6.70514784]
```

In []:

```
Y_predic = logisticRegr.predict(x_test)
Y_predic
```

Out[]:

[illegible]

0 1
Predicted label

In []:

```
print(classification_report(y_test,Y_predic))
```

	precision	recall	f1-score	support
0	0.85	1.00	0.92	809
1	1.00	0.03	0.07	148
accuracy			0.85	957
macro avg	0.92	0.52	0.49	957
weighted avg	0.87	0.85	0.79	957

Tingkat akurasi dari simulasi penerapan logistic regression di atas adalah 85%.