

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

In [2]: df = pd.read_csv("fraud_detection_train.csv")
df.head()
```

Out[2]:

	visit_id	kdkc	dati2	typeppk	jkpst	umur	jnspelsep	los	cmg	severitylevel	...	proc63_67	proc68_70	proc71_73	proc74_75	proc76_77	proc78_79	proc
0	1	1107	150	SB	P	64	2	0	F	0	...	0	0	0	0	0	0	
1	2	1303	200	C	L	45	1	9	E	3	...	0	0	0	0	0	0	
2	3	1114	172	B	P	34	2	0	Q	0	...	0	0	0	0	0	0	
3	4	601	90	SC	L	34	2	0	Q	0	...	0	0	0	0	0	0	
4	5	1006	130	B	L	27	2	0	F	0	...	0	0	0	0	0	0	

5 rows × 53 columns

```
In [3]: to_drop = ['visit_id', 'kdkc', 'dati2', 'typeppk', 'jkpst', 'cmg', 'diagprimer']

In [4]: df.drop(to_drop, inplace=True, axis=1)

In [5]: df.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200217 entries, 0 to 200216
Data columns (total 46 columns):
Column Non-Null Count Dtype
--- ---
0 umur 200217 non-null int64
1 jnspelsep 200217 non-null int64
2 los 200217 non-null int64
3 severitylevel 200217 non-null int64
4 dx2_a00_b99 200217 non-null int64
5 dx2_c00_d48 200217 non-null int64
6 dx2_d50_d89 200217 non-null int64
7 dx2_e00_e90 200217 non-null int64
8 dx2_f00_f99 200217 non-null int64
9 dx2_g00_g99 200217 non-null int64
10 dx2_h00_h59 200217 non-null int64
11 dx2_h60_h95 200217 non-null int64
12 dx2_i00_i99 200217 non-null int64
13 dx2_j00_j99 200217 non-null int64
14 dx2_koo_k93 200217 non-null int64
15 dx2_l00_l99 200217 non-null int64
16 dx2_m00_m99 200217 non-null int64
17 dx2_n00_n99 200217 non-null int64
18 dx2_o00_o99 200217 non-null int64
19 dx2_p00_p96 200217 non-null int64
20 dx2_q00_q99 200217 non-null int64
21 dx2_r00_r99 200217 non-null int64
22 dx2_s00_t98 200217 non-null int64
23 dx2_u00_u99 200217 non-null int64
24 dx2_v01_y98 200217 non-null int64
25 dx2_z00_z99 200217 non-null int64
26 proc00_13 200217 non-null int64
27 proc14_23 200217 non-null int64
28 proc24_27 200217 non-null int64
29 proc28_28 200217 non-null int64
30 proc29_31 200217 non-null int64
31 proc32_38 200217 non-null int64
32 proc39_45 200217 non-null int64
33 proc46_51 200217 non-null int64
34 proc52_57 200217 non-null int64
35 proc58_62 200217 non-null int64
36 proc63_67 200217 non-null int64
37 proc68_70 200217 non-null int64
38 proc71_73 200217 non-null int64
39 proc74_75 200217 non-null int64
40 proc76_77 200217 non-null int64
41 proc78_79 200217 non-null int64
42 proc80_99 200217 non-null int64
43 proce00_e99 200217 non-null int64
44 procv00_v89 200217 non-null int64
45 label 200217 non-null int64
dtypes: int64(46)
memory usage: 70.3 MB

```
In [6]: x = df.drop(["umur"], axis=1)
x.head(11)
```

Out[6]:

	jnspelsep	los	severitylevel	dx2_a00_b99	dx2_c00_d48	dx2_d50_d89	dx2_e00_e90	dx2_f00_f99	dx2_g00_g99	dx2_h00_h59	...	proc63_67	proc68_70	pr
0	2	0	0	0	0	0	0	0	0	0	...	0	0	
1	1	9	3	1	0	0	0	0	0	0	...	0	0	
2	2	0	0	0	0	0	0	0	0	0	...	0	0	
3	2	0	0	0	0	0	0	0	0	0	...	0	0	
4	2	0	0	0	0	0	0	0	0	0	...	0	0	
5	2	0	0	0	0	0	0	0	0	0	...	0	0	
6	1	3	1	0	0	0	0	0	0	0	...	0	0	
7	2	0	0	0	0	0	0	0	0	2	...	0	0	
8	2	0	0	0	0	0	0	0	0	0	...	0	0	
9	1	3	3	0	0	2	0	0	0	0	...	0	0	
10	2	0	0	0	0	0	0	0	0	0	...	0	0	

11 rows × 45 columns

```
In [7]: y = df["umur"]
y.head(11)
```

Out[7]:

```
0    64
1    45
2    34
3    34
4    27
5     0
6    73
7    64
8    21
9    44
10   79
Name: umur, dtype: int64
```

```
In [8]: from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

modelnb = GaussianNB()
```

```
In [9]: nbtrain = modelnb.fit(x, y)
df.head(11)
```

Out[9]:

	umur	jnspelsep	los	severitylevel	dx2_a00_b99	dx2_c00_d48	dx2_d50_d89	dx2_e00_e90	dx2_f00_f99	dx2_g00_g99	...	proc63_67	proc68_70	proc71_72
0	64	2	0	0	0	0	0	0	0	0	...	0	0	(
1	45	1	9	3	1	0	0	0	0	0	...	0	0	(
2	34	2	0	0	0	0	0	0	0	0	...	0	0	(
3	34	2	0	0	0	0	0	0	0	0	...	0	0	(
4	27	2	0	0	0	0	0	0	0	0	...	0	0	(
5	0	2	0	0	0	0	0	0	0	0	...	0	0	(
6	73	1	3	1	0	0	0	0	0	0	...	0	0	(
7	64	2	0	0	0	0	0	0	0	0	...	0	0	(
8	21	2	0	0	0	0	0	0	0	0	...	0	0	(
9	44	1	3	3	0	0	2	0	0	0	...	0	0	(
10	79	2	0	0	0	0	0	0	0	0	...	0	0	(

11 rows × 46 columns

In [10]:

x_test = df.drop(["umur"], axis=1)
x_test.head(11)

Out[10]:

	jnspelsep	los	severitylevel	dx2_a00_b99	dx2_c00_d48	dx2_d50_d89	dx2_e00_e90	dx2_f00_f99	dx2_g00_g99	dx2_h00_h59	...	proc63_67	proc68_70	pr
0	2	0	0	0	0	0	0	0	0	0 ...		0	0	
1	1	9	3	1	0	0	0	0	0	0 ...		0	0	
2	2	0	0	0	0	0	0	0	0	0 ...		0	0	
3	2	0	0	0	0	0	0	0	0	0 ...		0	0	
4	2	0	0	0	0	0	0	0	0	0 ...		0	0	
5	2	0	0	0	0	0	0	0	0	0 ...		0	0	
6	1	3	1	0	0	0	0	0	0	0 ...		0	0	
7	2	0	0	0	0	0	0	0	0	2 ...		0	0	
8	2	0	0	0	0	0	0	0	0	0 ...		0	0	
9	1	3	3	0	0	2	0	0	0	0 ...		0	0	
10	2	0	0	0	0	0	0	0	0	0 ...		0	0	

11 rows × 45 columns

In [11]:

y_uji = df["umur"]
y_uji.head(11)

Out[11]:

0 64
1 45
2 34
3 34
4 27
5 0
6 73
7 64
8 21
9 44
10 79
Name: umur, dtype: int64

In [12]:

Y_predict = nbtrain.predict(x_test)
print("Prediksi Naive Bayes : ",Y_predict)

Prediksi Naive Bayes : [105 90 105 ... 104 104 104]

In [13]:

from sklearn.metrics import accuracy_score
accuracy= accuracy_score(y_uji, Y_predict)
print("Akurasi Naive Bayes : ",accuracy)

Akurasi Naive Bayes : 0.01877962410784299

In [14]:

from sklearn.metrics import classification_report
print(classification_report(y_uji, Y_predict))

C:\Users\ROG\AppData\Roaming\Python\Python39\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: P
recision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\Users\ROG\AppData\Roaming\Python\Python39\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: P
recision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
C:\Users\ROG\AppData\Roaming\Python\Python39\site-packages\sklearn\metrics_classification.py:1334: UndefinedMetricWarning: P
recision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))

	precision	recall	f1-score	support
0	0.99	0.20	0.33	15856
1	0.13	0.00	0.01	3647
2	0.07	0.02	0.03	2824
3	0.00	0.00	0.00	2328
4	0.00	0.00	0.00	2141

In []: