

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

```
In [2]: df = pd.read_csv("case_cost_prediction_train.csv")
df.head(20)
```

Out[2]:

	row_id	tglpelayanan	kddati2	tkp	peserta	a	b	c	cb	d	...	kp	kt	ku	s	sa	sb	sc	sd	case	unit_cost
0	1	2014-04-01 00:00:00	332	40	179530	0	0	1	0	1	...	0	0	0	0	0	0	0	0	266	3.597440e+06
1	2	2016-11-01 00:00:00	54	40	104782	0	1	0	0	0	...	0	0	1	0	0	0	1	1	2453	4.951008e+06
2	3	2016-05-01 00:00:00	323	30	280645	0	0	1	0	0	...	0	0	0	0	0	0	0	0	1690	1.984208e+05
3	4	2018-11-01 00:00:00	318	40	178685	0	1	0	0	0	...	0	0	0	0	0	0	0	3	1321	4.008756e+06
4	5	2019-10-01 00:00:00	150	30	1199321	1	0	1	0	0	...	1	0	0	0	0	1	0	5	73056	3.072272e+05
5	6	2014-10-01 00:00:00	37	40	110377	0	1	0	0	0	...	0	0	0	0	0	0	1	0	1074	3.426614e+06
6	7	2020-05-01 00:00:00	379	40	105699	0	0	1	0	0	...	0	0	0	0	0	0	0	0	262	2.943943e+06
7	8	2015-09-01 00:00:00	110	40	152065	0	0	1	0	0	...	0	0	0	0	0	0	1	1	1102	3.971695e+06
8	9	2015-03-01 00:00:00	303	30	179081	0	1	0	0	1	...	0	0	0	0	0	0	0	0	5533	2.317746e+05
9	10	2017-11-01 00:00:00	49	30	227227	0	0	1	0	0	...	0	0	0	0	0	0	0	0	6426	1.903432e+05
10	11	2016-02-01 00:00:00	319	40	328163	0	2	1	0	0	...	0	0	0	0	0	0	1	0	4385	3.884721e+06
11	12	2019-09-01 00:00:00	89	40	203111	0	0	1	0	0	...	0	0	0	0	0	0	2	0	2457	3.509999e+06
12	13	2020-03-01 00:00:00	414	40	101559	0	0	1	0	0	...	0	0	0	0	0	0	0	0	462	3.406753e+06
13	14	2020-11-01 00:00:00	162	30	779562	0	0	2	0	0	...	1	0	0	0	0	0	1	1	24112	2.642148e+05
14	15	2014-04-01 00:00:00	198	30	595296	0	1	1	0	0	...	0	0	0	0	0	0	0	1	9340	1.985541e+05
15	16	2018-05-01 00:00:00	349	40	179578	0	0	1	0	0	...	0	0	0	0	0	0	0	0	480	3.299373e+06
16	17	2020-01-01 00:00:00	256	30	112851	0	0	1	0	1	...	0	0	0	0	0	0	0	0	1350	2.598507e+05
17	18	2016-03-01 00:00:00	424	40	129935	0	0	1	0	0	...	0	0	0	0	0	0	0	0	671	3.339520e+06
18	19	2019-11-01 00:00:00	64	30	360938	0	1	1	0	0	...	0	0	0	0	0	0	1	2	21074	2.322047e+05
19	20	2015-09-01 00:00:00	97	30	265311	0	1	1	0	0	...	0	0	0	0	0	0	3	0	13680	2.400168e+05

20 rows × 36 columns

```
In [3]: to_drop = ['unit_cost', 'peserta',
                  'row_id',
                  'tglpelayanan',
                  'kddati2',]
```

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

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

Out[5]:

	tkp	a	b	c	cb	d	ds	gd	hd	il	...	ko	kp	kt	ku	s	sa	sb	sc	sd	case
0	40	0	0	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	266
1	40	0	1	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	1	1	2453
2	30	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1690
3	40	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	3	1321
4	30	1	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	1	0	5	73056

5 rows × 31 columns

In [6]: df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57971 entries, 0 to 57970
Data columns (total 31 columns):
#   Column      Non-Null Count  Dtype
---  -
0   tkp         57971 non-null  int64
1   a           57971 non-null  int64
2   b           57971 non-null  int64
3   c           57971 non-null  int64
4   cb          57971 non-null  int64
5   d           57971 non-null  int64
6   ds          57971 non-null  int64
7   gd          57971 non-null  int64
8   hd          57971 non-null  int64
9   i1          57971 non-null  int64
10  i2          57971 non-null  int64
11  i3          57971 non-null  int64
12  i4          57971 non-null  int64
13  kb          57971 non-null  int64
14  kc          57971 non-null  int64
15  kg          57971 non-null  int64
16  ki          57971 non-null  int64
17  kj          57971 non-null  int64
18  kk          57971 non-null  int64
19  kl          57971 non-null  int64
20  km          57971 non-null  int64
21  ko          57971 non-null  int64
22  kp          57971 non-null  int64
23  kt          57971 non-null  int64
24  ku          57971 non-null  int64
25  s           57971 non-null  int64
26  sa          57971 non-null  int64
27  sb          57971 non-null  int64
28  sc          57971 non-null  int64
29  sd          57971 non-null  int64
30  case        57971 non-null  int64
dtypes: int64(31)
memory usage: 13.7 MB

```

In [7]: x = df.drop(["tkp"], axis=1)
x.head(11)

Out[7]:

	a	b	c	cb	d	ds	gd	hd	i1	i2	...	ko	kp	kt	ku	s	sa	sb	sc	sd	case
0	0	0	1	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	266
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	1	1	2453
2	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1690
3	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	3	1321
4	1	0	1	0	0	0	0	0	0	0	...	0	1	0	0	0	0	1	0	5	73056
5	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	1074
6	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	262
7	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	1	1102
8	0	1	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	5533
9	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	6426
10	0	2	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	4385

11 rows × 30 columns

In [8]: y = df["tkp"]
y.head(11)

Out[8]:

0	40
1	40
2	30
3	40
4	30
5	40
6	40
7	40
8	30
9	30
10	40

Name: tkp, dtype: int64

```
In [9]: from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

modelnb = GaussianNB()
```

```
In [10]: nbtrain = modelnb.fit(x, y)
df.head(11)
```

```
Out[10]:
```

	tkp	a	b	c	cb	d	ds	gd	hd	l1	...	ko	kp	kt	ku	s	sa	sb	sc	sd	case
0	40	0	0	1	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	266
1	40	0	1	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	1	1	2453
2	30	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1690
3	40	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	3	1321
4	30	1	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	1	0	5	73056
5	40	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	1074
6	40	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	262
7	40	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	1	1102
8	30	0	1	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	5533
9	30	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	6426
10	40	0	2	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	4385

11 rows × 31 columns

```
In [11]: x_test = df.drop(["tkp"], axis=1)
x_test.head(11)
```

```
Out[11]:
```

	a	b	c	cb	d	ds	gd	hd	l1	l2	...	ko	kp	kt	ku	s	sa	sb	sc	sd	case
0	0	0	1	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	266
1	0	1	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	1	1	2453
2	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1690
3	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	3	1321
4	1	0	1	0	0	0	0	0	0	0	...	0	1	0	0	0	0	1	0	5	73056
5	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	1074
6	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	262
7	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	1	1102
8	0	1	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	5533
9	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	6426
10	0	2	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	4385

11 rows × 30 columns

```
In [12]: y_uji = df["tkp"]
y_uji.head(11)
```

```
Out[12]:
```

0	40
1	40
2	30
3	40
4	30
5	40
6	40
7	40
8	30
9	30
10	40

Name: tkp, dtype: int64

```
In [13]: Y_predict = nbtrain.predict(x_test)
print("Prediksi Naive Bayes : ",Y_predict)
```

Prediksi Naive Bayes : [40 40 40 ... 40 40 30]

```
In [14]: from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_uji, Y_predict)
print("Akurasi Naive Bayes : ",accuracy)
```

Akurasi Naive Bayes : 0.6359559089889772

```
In [15]: from sklearn.metrics import classification_report
print(classification_report(y_uji, Y_predict))
```

	precision	recall	f1-score	support
30	0.88	0.32	0.47	29043
40	0.58	0.96	0.72	28928
accuracy			0.64	57971
macro avg	0.73	0.64	0.59	57971
weighted avg	0.73	0.64	0.59	57971