## Import library

```
import pandas as pd
import statsmodels.formula.api as smf
import numpy as np
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

ds = pd.read_csv("D:/STATISTIKA\BANK LOAN.csv")
ds
```

```
     age  ed  employ  address  income  debtinc   creddebt   othdebt
default
0     41   3      17       12     176      9.3  11.359392  5.008608
1
1     27   1      10        6      31     17.3   1.362202  4.000798
0
2     40   1      15       14      55      5.5   0.856075  2.168925
0
3     41   1      15       14     120      2.9   2.658720  0.821280
0
4     24   2       2        0      28     17.3   1.787436  3.056564
1
..   ...  ..     ...      ...     ...      ...        ...       ...
...
695   36   2       6       15      27      4.6   0.262062  0.979938
1
696   29   2       6        4      21     11.5   0.369495  2.045505
0
697   33   1      15        3      32      7.6   0.491264  1.940736
0
698   45   1      19       22      77      8.4   2.302608  4.165392
0
699   37   1      12       14      44     14.7   2.994684  3.473316
0

[700 rows x 9 columns]
```

## Menampilkan eksplanatory data analitik dari data BANK LOAN

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
```

```
 0   age       700 non-null    int64
 1   ed        700 non-null    int64
 2   employ    700 non-null    int64
 3   address   700 non-null    int64
 4   income    700 non-null    int64
 5   debtinc   700 non-null    float64
 6   creddebt  700 non-null    float64
 7   othdebt   700 non-null    float64
 8   default   700 non-null    int64
dtypes: float64(3), int64(6)
memory usage: 49.3 KB
```

```
ds['age'] = pd.cut(ds['age'], bins=[0,28,40,150],
labels=['1','2','3'])
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       700 non-null    category
 1   ed        700 non-null    int64
 2   employ    700 non-null    int64
 3   address   700 non-null    int64
 4   income    700 non-null    int64
 5   debtinc   700 non-null    float64
 6   creddebt  700 non-null    float64
 7   othdebt   700 non-null    float64
 8   default   700 non-null    int64
dtypes: category(1), float64(3), int64(5)
memory usage: 44.7 KB
```

```
ds.head(10)
```

```
  age  ed  employ  address  income  debtinc   creddebt    othdebt
default
0   3   3      17       12     176      9.3  11.359392   5.008608
1
1   1   1      10        6      31     17.3   1.362202   4.000798
0
2   2   1      15       14      55      5.5   0.856075   2.168925
0
3   3   1      15       14     120      2.9   2.658720   0.821280
0
4   1   2       2        0      28     17.3   1.787436   3.056564
1
5   3   2       5        5      25     10.2   0.392700   2.157300
0
6   2   1      20        9      67     30.6   3.833874  16.668126
```

```
0
7   3   1      12       11      38      3.6   0.128592   1.239408
0
8   1   1       3        4      19     24.4   1.358348   3.277652
1
9   2   1       0       13      25     19.7   2.777700   2.147300
0

resiko_model = smf.logit(formula='default ~ age + employ + address +
debtinc + creddebt + othdebt',
                         data = ds).fit()
resiko_model.summary()

Optimization terminated successfully.
         Current function value: 0.395614
         Iterations 7

<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results

==============================================================================
========
Dep. Variable:                 default   No. Observations:
700
Model:                           Logit   Df Residuals:
692
Method:                            MLE   Df Model:
7
Date:                Thu, 28 Mar 2024   Pseudo R-squ.:
0.3114
Time:                        09:42:06   Log-Likelihood:
-276.93
converged:                       True   LL-Null:
-402.18
Covariance Type:            nonrobust   LLR p-value:
2.164e-50
==============================================================================
========
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept     -0.7984      0.271     -2.950      0.003      -1.329
-0.268
age[T.2]       0.1782      0.269      0.662      0.508      -0.349
0.706
age[T.3]       0.5988      0.381      1.571      0.116      -0.148
1.346
employ        -0.2596      0.032     -8.105      0.000      -0.322
```

```
-0.197
address        -0.0978      0.023     -4.336      0.000       -0.142
-0.054
debtinc         0.0849      0.022      3.842      0.000        0.042
0.128
creddebt        0.5641      0.089      6.310      0.000        0.389
0.739
othdebt         0.0231      0.057      0.405      0.686       -0.089
0.135
==============================================================================
========
"""

resiko_model = smf.logit(formula='default ~ employ + address + debtinc
+ creddebt',
                         data = ds).fit()
resiko_model.summary()

Optimization terminated successfully.
         Current function value: 0.397665
         Iterations 7

<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results

==============================================================================
========
Dep. Variable:                default   No. Observations:
700
Model:                          Logit   Df Residuals:
695
Method:                           MLE   Df Model:
4
Date:                Thu, 28 Mar 2024   Pseudo R-squ.:
0.3079
Time:                        09:42:06   Log-Likelihood:
-278.37
converged:                       True   LL-Null:
-402.18
Covariance Type:            nonrobust   LLR p-value:
2.106e-52
==============================================================================
========
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept     -0.7911      0.252     -3.145      0.002      -1.284
-0.298
```

```
employ         -0.2426      0.028     -8.646      0.000      -0.298
-0.188
address        -0.0812      0.020     -4.145      0.000      -0.120
-0.043
debtinc         0.0883      0.019      4.760      0.000       0.052
0.125
creddebt        0.5730      0.087      6.566      0.000       0.402
0.744
==============================================================================
========
"""
```

Kedua hasil OLS di atas adalah tampilan summary dari logistic regresion menggunakan OLS. Dimana di logit model kedua dengan tanpa atribut umur menghasilkan nilai yang sedikit lebih tinggi.

```
conf = resiko_model.conf_int()
conf['OR'] = resiko_model.params
conf.columns = ['2.5%', '97.5%', 'OR']
print(np.exp(conf))
```

```
                2.5%     97.5%        OR
Intercept   0.276905  0.742263  0.453361
employ      0.742606  0.828941  0.784587
address     0.887222  0.958073  0.921967
debtinc     1.053295  1.132707  1.092279
creddebt    1.494738  2.104432  1.773577
```

Hasil output merupakan nilai interval kepercayaan (confidence interval) dan nilai Odds Ratio (OR) dari model logit. Besaran nilai merupakan ukuran dari seberapa besar kemungkinan perubahan pada variabel independen mempengaruhi peluang kejadian pada variabel dependen.

```
predicted_value1 = resiko_model.predict()
threshold =  0.5 #persentase
predicted_class1 = np.zeros(predicted_value1.shape)
predicted_class1[predicted_value1>threshold]=1
cm1 = confusion_matrix(ds['default'], predicted_class1)
print('confusion matrix  : \n', cm1)
```

```
confusion matrix  :
 [[478  39]
 [ 91  92]]
```

Menampilkan hasil confusion matrix dari model logit, hal ini dilakukan untuk mengidentifikasi metrik evaluasi

```
sensitivity = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Sensitivity : ', sensitivity)
specifity = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Specifity : ', specifity)

Sensitivity :  0.5027322404371585
Specifity :  0.9245647969052224
```

Model memiliki tingkat sensitivitas sekitar 50.27%, yang mengindikasikan bahwa hanya sekitar separuh dari kasus positif aktual yang diprediksi dengan benar oleh model. Di sisi lain, tingkat spesifisitas model sekitar 92.46%, yang menunjukkan bahwa sebagian besar kasus negatif aktual diprediksi dengan benar oleh model.

```
print(classification_report(ds['default'], predicted_class1))

              precision    recall  f1-score   support

           0       0.84      0.92      0.88       517
           1       0.70      0.50      0.59       183

    accuracy                           0.81       700
   macro avg       0.77      0.71      0.73       700
weighted avg       0.80      0.81      0.80       700
```

Terakhir menampilkan hasil confusion matrix dari model logit