```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
```

```
df = pd.read_csv("bank-loan.csv")
df
```

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
0	41	3	17	12	176	9.3	11.359392	5.008608	1.0
1	27	1	10	6	31	17.3	1.362202	4.000798	0.0
2	40	1	15	14	55	5.5	0.856075	2.168925	0.0
3	41	1	15	14	120	2.9	2.658720	0.821280	0.0
4	24	2	2	0	28	17.3	1.787436	3.056564	1.0
...	...	...	...	...	...	...	...	...	...
845	34	1	12	15	32	2.7	0.239328	0.624672	NaN
846	32	2	12	11	116	5.7	4.026708	2.585292	NaN
847	48	1	13	11	38	10.8	0.722304	3.381696	NaN
848	35	2	1	11	24	7.8	0.417456	1.454544	NaN
849	37	1	20	13	41	12.9	0.899130	4.389870	NaN

850 rows × 9 columns

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

```
age           0
ed            0
employ        0
address       0
income        0
debtinc       0
creddebt      0
othdebt       0
default     150
dtype: int64
```

```
df
```

	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
0	41	3	17	12	176	9.3	11.359392	5.008608	1.0
1	27	1	10	6	31	17.3	1.362202	4.000798	0.0
2	40	1	15	14	55	5.5	0.856075	2.168925	0.0
3	41	1	15	14	120	2.9	2.658720	0.821280	0.0
4	24	2	2	0	28	17.3	1.787436	3.056564	1.0
...	...	...	...	...	...	...	...	...	...
845	34	1	12	15	32	2.7	0.239328	0.624672	NaN
846	32	2	12	11	116	5.7	4.026708	2.585292	NaN
847	48	1	13	11	38	10.8	0.722304	3.381696	NaN
848	35	2	1	11	24	7.8	0.417456	1.454544	NaN

```
df.default.value_counts() #mode=0 value - replace by 0
```

```
0.0    517
1.0    183
Name: default, dtype: int64
```

```
df['default'] = df['default'].fillna(0.0)
```

```
import seaborn as sns
```

```
df.columns
```

```
Index(['age', 'ed', 'employ', 'address', 'income', 'debtinc', 'creddebt',
       'othdebt', 'default'],
      dtype='object')
```

```
#split dataset in features and target variable
feature = ['age', 'ed', 'employ', 'address', 'income', 'debtinc', 'creddebt','othdebt',]
X = df[feature] # Features
y = df.default # Target variable
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7)
```

```
X_train
```

	age	ed	employ	address	income	debtinc	creddebt	othdebt
98	29	2	4	9	19	6.3	0.681093	0.515907
276	26	2	2	6	24	13.6	1.576512	1.687488
233	35	2	5	1	33	2.8	0.103488	0.820512
131	33	1	5	10	18	12.9	0.984528	1.337472
817	38	2	0	18	21	4.6	0.612444	0.353556
...	...	...	...	...	...	...	...	...
269	39	1	19	16	45	3.7	0.915750	0.749250
351	22	1	1	3	17	18.6	0.806310	2.355690
128	21	1	5	1	25	9.0	0.366750	1.883250
624	29	1	6	4	46	5.4	1.169964	1.314036

```
logreg = LogisticRegression()
```

595 rows × 8 columns

```
# fit the model with data
logreg.fit(X_train,y_train)
```

```
LogisticRegression()
```

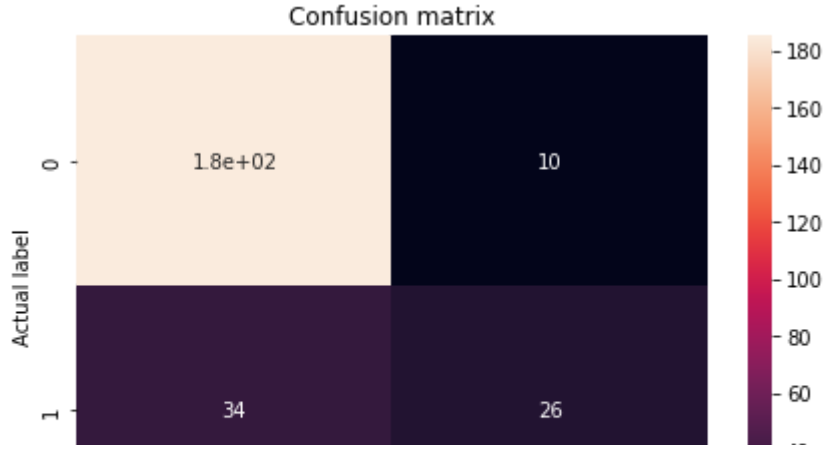
```
#
y_pred=logreg.predict(X_test)
```

```
# import the metrics class
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

```
array([[185,  10],
       [ 34,  26]])
```

```
import seaborn as sns
sns.heatmap(cnf_matrix, annot=True)
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

```
Text(0.5, 15.0, 'Predicted label')
```




```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred))
print("Recall:",metrics.recall_score(y_test, y_pred))
```

```
Accuracy: 0.8274509803921568
Precision: 0.7222222222222222
Recall: 0.43333333333333335
```