```
import pandas as pd
import numpy as np
import tensorflow as tf
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler


import xgboost as xgb
from sklearn.linear_model import LogisticRegression
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import auc
from sklearn.metrics import roc_curve
sns.set()



from google.colab import drive
drive.mount("/content/drive")

data = pd.read_csv("/content/drive/My Drive/dataset/kredit.csv")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
data
```

1 to 25 of 31 entries Filter

index	id_pelanggan	usia	jk	pendapatan	jmlh_kartu_kredit	pengeluaran	respon
0	1	40	1	72000000	3	6000000	1
1	2	28	1	36000000	1	4500000	0
2	3	31	0	60000000	2	5000000	0
3	4	28	1	36000000	1	3000000	1
4	5	38	0	72000000	2	5000000	1
5	6	28	1	58000000	2	6200000	1
6	7	40	0	36000000	1	4000000	0
7	8	38	1	72000000	4	10000000	1
8	9	31	0	70000000	3	8000000	0
9	10	41	1	60000000	1	5300000	0
10	11	44	0	36000000	1	4500000	1
11	12	28	1	58000000	2	5000000	0
12	13	41	0	72000000	4	12000000	1
13	14	38	1	68000000	3	9000000	0
14	15	40	0	45000000	3	8000000	1
15	16	37	1	36000000	2	5400000	1
16	17	28	0	60000000	1	3000000	1
17	18	38	0	47000000	1	4000000	0
18	19	40	1	42000000	2	5200000	0
19	20	32	1	36000000	3	8000000	1
20	21	32	0	65000000	3	11000000	0
21	22	32	0	72000000	1	5000000	1
22	23	36	1	42000000	2	8000000	0
23	24	40	1	65000000	2	6100000	0
24	25	36	0	36000000	1	4500000	1

Show 25 per page 1 2

Like what you see? Visit the data table notebook to learn more about interactive tables.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id_pelanggan       31 non-null     int64
```

```
 1   usia              31 non-null     int64
 2   jk                31 non-null     object
 3   pendapatan        31 non-null     int64
 4   jmlh_kartu_kredit  31 non-null     int64
 5   pengeluaran       31 non-null     int64
 6   respon            31 non-null     int64
dtypes: int64(6), object(1)
memory usage: 1.8+ KB
```

```
data.describe()
```

	id_pelanggan	usia	pendapatan	jmlh_kartu_kredit	pengeluar
count	31.000000	31.000000	3.100000e+01	31.000000	3.100000e+
mean	16.000000	34.354839	5.506452e+07	1.903226	5.903226e+
std	9.092121	5.588968	1.364047e+07	0.943569	2.263402e+
min	1.000000	24.000000	3.600000e+07	1.000000	3.000000e+
25%	8.500000	29.500000	4.200000e+07	1.000000	4.500000e+
50%	16.000000	36.000000	5.800000e+07	2.000000	5.000000e+
75%	23.500000	40.000000	6.650000e+07	2.500000	7.100000e+

```
data.isna().sum()
```

```
id_pelanggan         0
usia                 0
jk                   0
pendapatan           0
jmlh_kartu_kredit    0
pengeluaran          0
respon               0
dtype: int64
```

```
data['jk'].value_counts()
```

```
P    17
L    14
Name: jk, dtype: int64
```

```
data['respon'].value_counts()
```

```
1    16
0    15
Name: respon, dtype: int64
```

```
data.dtypes
```

```
id_pelanggan          int64
usia                  int64
jk                   object
pendapatan            int64
jmlh_kartu_kredit     int64
pengeluaran           int64
respon                int64
dtype: object
```

```
for col in data.columns:
    if data[col].dtypes == 'object':
        print(col, data[col].unique())
```

```
jk ['P' 'L']
```

```
cols_to_label=[]
for i in data.columns:
    if data[i].dtypes == 'O':
        cols_to_label.append(i)
```

```
cols_to_label
```

```
['jk']
```

```
data[cols_to_label] = data[cols_to_label].apply(LabelEncoder().fit_transform)
```

```
data.head()
```

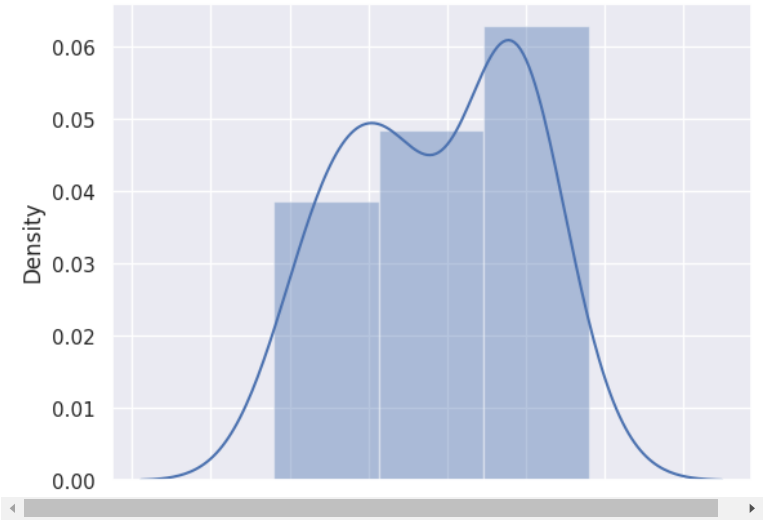
	id_pelanggan	usia	jk	pendapatan	jmlh_kartu_kredit	pengeluaran	res
0	1	40	1	72000000	3	6000000	
1	2	28	1	36000000	1	4500000	
2	3	31	0	60000000	2	5000000	
3	4	28	1	36000000	1	3000000	

```
data.isnull().sum()
```

```
id_pelanggan         0
usia                 0
jk                   0
pendapatan           0
jmlh_kartu_kredit    0
pengeluaran          0
respon               0
dtype: int64
```

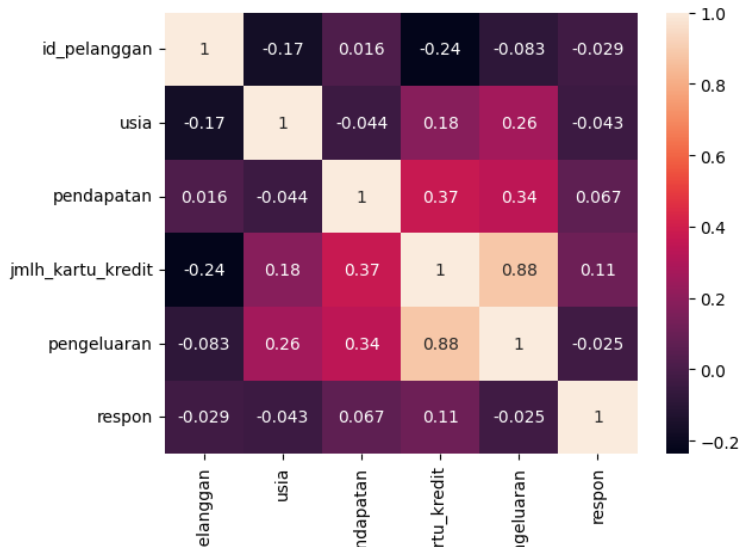
```
sns.distplot(data['usia'])
```

```
<ipython-input-49-153e0390af28>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0

Please adapt your code to use either `displot` (a figure-level function wi
similar flexibility) or `histplot` (an axes-level function for histograms)

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['usia'])
<Axes: xlabel='usia', ylabel='Density'>
```




```
import seaborn as sns
sns.heatmap(data.corr(),annot=True)
```

```
<ipython-input-33-53fe516a88e7>:2: FutureWarning: The default value of num
  sns.heatmap(data.corr(),annot=True)
<Axes: >
```




```
X_train, X_test, y_train, y_test = train_test_split(data.drop('respon', axis=1), data['respon'], test_size=0.2)
```

```
model1 = LogisticRegression(random_state = 20).fit(X_train, y_train)
preds = model1.predict(X_test)
print(f'The accuracy score of Logistic Regression model is: {accuracy_score(preds, y_test)}')
```

```
The accuracy score of Logistic Regression model is: 0.42857142857142855
```

```
model2 = XGBClassifier().fit(X_train, y_train)
preds = model2.predict(X_test)
print(f'The accuracy score of XGBClassifier model is: {accuracy_score(preds, y_test)}')
```

```
The accuracy score of XGBClassifier model is: 0.42857142857142855
```