# Kelompok 3 C2

1. 2101310 AKMAL ZULKIFLI
2. 2205297 NABILLA ASSYIFA RAMADHANI
3. 2205410 RADITYA ADHA RAHMAN
4. 2106330 WILDAN WALILI DARDJAT
5. 2201017 ILHAM AKBAR

Data NPL untuk mendeteksi nasabah yang kreditnya akan macet. Field yang akan diprediksi adalah "flag_kredit_macet".

## IMPORT LIBRARY

In [1]:
```
# display
%matplotlib inline
# numerik
import numpy as np
# analisis
import pandas as pd
# plot
import matplotlib.pyplot as plt
# visualisasi
import seaborn as sns
sns.set()
warnings.filterwarnings("ignore")
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from xgboost import XGBClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
```

## READ FILE

In [2]:
```
data=pd.read_csv("npl_train.txt")
data.head(3)
```

Out[2]:

| | X | jumlah_kartu | outstanding | limit_kredit | tagihan | total_pemakaian_tunai | total_pemakaian_retail | sisa_tagihan_tidak_terbayar | kode_cabang | rasio_pembayaran | ... | flag_kredit_macet | jumlah_tahun |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 36150 | 70000000.0 | 23437.0 | 0.0 | 94.0 | 26323.0 | I | 102.19 | ... | 0 | |
| **1** | 2 | 2 | 268691 | 100000000.0 | 254564.0 | 0.0 | 1012.0 | 0.0 | A | 0.00 | ... | 0 | |
| **2** | 3 | 3 | 6769149 | 280000000.0 | 4159779.0 | 0.0 | 0.0 | 0.0 | A | 100.00 | ... | 0 | |

3 rows × 24 columns

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15493 entries, 0 to 15492
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   X                           15493 non-null  int64  
 1   jumlah_kartu                15493 non-null  int64  
 2   outstanding                 15493 non-null  int64  
 3   limit_kredit                15493 non-null  float64
 4   tagihan                     15493 non-null  float64
 5   total_pemakaian_tunai       15493 non-null  float64
 6   total_pemakaian_retail      15493 non-null  float64
 7   sisa_tagihan_tidak_terbayar 15493 non-null  float64
 8   kode_cabang                 15393 non-null  object 
 9   rasio_pembayaran            15493 non-null  float64
 10  persentasi_overlimit        15493 non-null  float64
 11  rasio_pembayaran_3bulan     15493 non-null  float64
 12  rasio_pembayaran_6bulan     15493 non-null  float64
 13  skor_delikuensi             15493 non-null  int64  
 14  flag_kredit_macet           15493 non-null  int64  
 15  jumlah_tahun_sejak_pembukaan_kredit  15493 non-null  float64
 16  total_pemakaian             15493 non-null  float64
 17  sisa_tagihan_per_jumlah_kartu  15493 non-null  float64
 18  sisa_tagihan_per_limit      15493 non-null  float64
 19  total_pemakaian_per_limit   15493 non-null  float64
 20  pemakaian_3bln_per_limit    15493 non-null  float64
 21  pemakaian_6bln_per_limit    15493 non-null  float64
 22  utilisasi_3bulan            15493 non-null  float64
 23  utilisasi_6bulan            15493 non-null  float64
dtypes: float64(18), int64(5), object(1)
memory usage: 2.8+ MB
```

## Menghapus Kode Cabang

kode_cabang dan X dihapus karena kedua tidak memiliki katan dengan hasil akhir.

In [4]:
```
data = data.drop(["kode_cabang","X"], axis=1)
```

In [5]:
```
data["flag_kredit_macet"].value_counts()
```

Out[5]:
```
0   14134
1    1359
Name: flag_kredit_macet, dtype: int64
```

## SPLIT DATA

In [6]:
```
from sklearn.model_selection import train_test_split
```

In [7]:
```
data_train, data_test=train_test_split(data,test_size=0.4,random_state=1)
```

In [8]:
```
data_train.head(5)
```

Out[8]:

| | jumlah_kartu | outstanding | limit_kredit | tagihan | total_pemakaian_tunai | total_pemakaian_retail | sisa_tagihan_tidak_terbayar | rasio_pembayaran | persentasi_overlimit | rasio_pembayaran_3bulan |
|---|---|---|---|---|---|---|---|---|---|---|
| **10502** | 4 | 24962677 | 230000000.0 | 22575640.0 | 0.0 | 2957906.0 | 25136225.0 | 0.0 | 8.62 | 7.88 |
| **14010** | 2 | 7474329 | 70000000.0 | 2901653.0 | 0.0 | 0.0 | 2133750.0 | 60.9 | 2.07 | 83.90 |
| **5089** | 3 | 596113 | 44000000.0 | 530033.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 | 18.10 |
| **7683** | 3 | 4853906 | 100000000.0 | 4748881.0 | 0.0 | 1130000.0 | 4746073.0 | 23.1 | 0.00 | 30.50 |
| **5692** | 2 | 76966 | 80000000.0 | 78000.0 | 0.0 | 196.0 | 0.0 | 0.0 | 0.00 | 25.20 |

5 rows × 22 columns

In [9]:
```
data_test=data_test.drop("flag_kredit_macet", 1)
```

In [10]:
```
data_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9295 entries, 10502 to 13349
Data columns (total 22 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   jumlah_kartu                9295 non-null   int64  
 1   outstanding                 9295 non-null   int64  
 2   limit_kredit                9295 non-null   float64
 3   tagihan                     9295 non-null   float64
 4   total_pemakaian_tunai       9295 non-null   float64
 5   total_pemakaian_retail      9295 non-null   float64
 6   sisa_tagihan_tidak_terbayar 9295 non-null   float64
 7   rasio_pembayaran            9295 non-null   float64
 8   persentasi_overlimit        9295 non-null   float64
 9   rasio_pembayaran_3bulan     9295 non-null   float64
 10  rasio_pembayaran_6bulan     9295 non-null   float64
 11  skor_delikuensi             9295 non-null   int64  
 12  flag_kredit_macet           9295 non-null   int64  
 13  jumlah_tahun_sejak_pembukaan_kredit  9295 non-null   float64
 14  total_pemakaian             9295 non-null   float64
 15  sisa_tagihan_per_jumlah_kartu  9295 non-null   float64
 16  sisa_tagihan_per_limit      9295 non-null   float64
 17  total_pemakaian_per_limit   9295 non-null   float64
 18  pemakaian_3bln_per_limit    9295 non-null   float64
 19  pemakaian_6bln_per_limit    9295 non-null   float64
 20  utilisasi_3bulan            9295 non-null   float64
 21  utilisasi_6bulan            9295 non-null   float64
dtypes: float64(18), int64(4)
memory usage: 1.6 MB
```

## Kelas Target

In [11]:
```
X=data_train.drop("flag_kredit_macet",1)
```

In [12]:
```
X.head(5)
```

Out[12]:

| | jumlah_kartu | outstanding | limit_kredit | tagihan | total_pemakaian_tunai | total_pemakaian_retail | sisa_tagihan_tidak_terbayar | rasio_pembayaran | persentasi_overlimit | rasio_pembayaran_3bulan |
|---|---|---|---|---|---|---|---|---|---|---|
| **10502** | 4 | 24962677 | 230000000.0 | 22575640.0 | 0.0 | 2957906.0 | 25136225.0 | 0.0 | 8.62 | 7.88 |
| **14010** | 2 | 7474329 | 70000000.0 | 2901653.0 | 0.0 | 0.0 | 2133750.0 | 60.9 | 2.07 | 83.90 |
| **5089** | 3 | 596113 | 44000000.0 | 530033.0 | 0.0 | 0.0 | 0.0 | 0.00 | 0.00 | 18.10 |
| **7683** | 3 | 4853906 | 100000000.0 | 4748881.0 | 0.0 | 1130000.0 | 4746073.0 | 23.1 | 0.00 | 30.50 |
| **5692** | 2 | 76966 | 80000000.0 | 78000.0 | 0.0 | 196.0 | 0.0 | 0.0 | 0.00 | 25.20 |

5 rows × 21 columns

In [13]:
```
y=data_train[["flag_kredit_macet"]]
```

In [14]:
```
y.head(5)
```

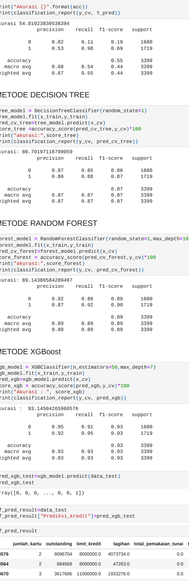
Out[14]:

| | flag_kredit_macet |
|---|---|
| **10502** | 0 |
| **14010** | 0 |
| **5089** | 1 |
| **7683** | 0 |
| **5692** | 1 |

## DATA IMBALANCE

In [15]:
```
plt.figure(figsize=(15,8))
plt.subplot(1,2,1)
y.value_counts().plot.pie(autopct='%1.1f%%')
centre=plt.Circle((0,0),0.7,fc='white')
fig=plt.gcf()
fig.gca().add_artist(centre)
```

Out[15]:
```
<matplotlib.patches.Circle at 0x29bef0bc110>
```




In [16]:
```
from imblearn.over_sampling import SMOTE
over=SMOTE()
X_over,y_over=over.fit_resample(X,y)
```

In [17]:
```
x_train,x_cv,y_train,y_cv=train_test_split(X,y,test_size=0.2,random_state=1)
```

In [18]:
```
plt.figure(figsize=(15,8))
plt.subplot(1,2,1)
y.value_counts().plot.pie(autopct='%1.1f%%')
centre=plt.Circle((0,0),0.7,fc='white')
fig=plt.gcf()
fig.gca().add_artist(centre)
```

Out[18]:
```
<matplotlib.patches.Circle at 0x229bfa14110>
```


## METODE NAIVE BAYES

In [19]:
```
nb_model = GaussianNB()
nb_model.fit(x_train, y_train)
Y_pred = nb_model.predict(x_cv)
acc = accuracy_score(y_cv, Y_pred)*100

print("Akurasi {}".format(acc))
print(classification_report(y_cv, Y_pred))
```

```
Akurasi 54.81623836583834
              precision    recall  f1-score   support

           0       0.82      0.11      0.19      1680
           1       0.53      0.98      0.69      1719

    accuracy                           0.55      3399
   macro avg       0.68      0.54      0.44      3399
weighted avg       0.67      0.55      0.44      3399
```

## METODE DECISION TREE

In [20]:
```
tree_model = DecisionTreeClassifier(random_state=1)
tree_model.fit(x_train,y_train)
pred_cv_tree=tree_model.predict(x_cv)
score_tree = accuracy_score(pred_cv_tree,y_cv)*100
print('Akurasi',score_tree)
print(classification_report(y_cv, pred_cv_tree))
```

```
Akurasi: 86.70197116799059
              precision    recall  f1-score   support

           0       0.87      0.85      0.86      1680
           1       0.86      0.88      0.87      1719

    accuracy                           0.87      3399
   macro avg       0.87      0.87      0.87      3399
weighted avg       0.87      0.87      0.87      3399
```

## METODE RANDOM FOREST

In [21]:
```
forest_model = RandomForestClassifier(random_state=1,max_depth=10,n_estimators=50)
forest_model.fit(x_train,y_train)
pred_cv_forest=forest_model.predict(x_cv)
score_forest = accuracy_score(pred_cv_forest,y_cv)*100
print('Akurasi:',score_forest)
print(classification_report(y_cv, pred_cv_forest))
```

```
Akurasi: 89.14386584289497
              precision    recall  f1-score   support

           0       0.92      0.86      0.89      1680
           1       0.87      0.92      0.90      1719

    accuracy                           0.89      3399
   macro avg       0.89      0.89      0.89      3399
weighted avg       0.89      0.89      0.89      3399
```

## METODE XGBoost

In [22]:
```
xgb_model = XGBClassifier(n_estimators=50,max_depth=7)
xgb_model.fit(x_train,y_train)
pred_xgb=xgb_model.predict(x_cv)
score_xgb = accuracy_score(pred_xgb,y_cv)*100
print('Akurasi : ', score_xgb)
print(classification_report(y_cv, pred_xgb))
```

```
Akurasi :  93.14504265960576
              precision    recall  f1-score   support

           0       0.95      0.91      0.93      1680
           1       0.92      0.95      0.93      1719

    accuracy                           0.93      3399
   macro avg       0.93      0.93      0.93      3399
weighted avg       0.93      0.93      0.93      3399
```

In [23]:
```
pred_xgb_test=xgb_model.predict(data_test)
pred_xgb_test
```

Out[23]:
```
array([0, 0, 0, ..., 0, 0, 1])
```

In [24]:
```
df_pred_result=data_test
df_pred_result["Prediksi_kredit"]=pred_xgb_test
```

In [25]:
```
df_pred_result
```

Out[25]:

| | jumlah_kartu | outstanding | limit_kredit | tagihan | total_pemakaian_tunai | total_pemakaian_retail | sisa_tagihan_tidak_terbayar | rasio_pembayaran | persentasi_overlimit | rasio_pembayaran_3bulan |
|---|---|---|---|---|---|---|---|---|---|---|
| **3576** | 2 | 8096704 | 80000000.0 | 4073734.0 | 0.0 | 1903325.0 | 5213856.0 | 65.5 | 1.24 | 37.90 |
| **8564** | 2 | 684568 | 80000000.0 | 472818.0 | 0.0 | 538655.0 | 0.0 | 100.0 | 0.00 | 100.00 |
| **13670** | 3 | 3617686 | 110000000.0 | 1933276.0 | 0.0 | 5860000.0 | 3155317.0 | 100.0 | 0.00 | 100.00 |
| **13243** | 2 | 3119979 | 30000000.0 | 3007042.0 | 0.0 | 100000.0 | 3097253.0 | 10.1 | 3.84 | 12.40 |
| **804** | 2 | 139723 | 40000000.0 | 148961.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.00 | 39.46 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8367** | 4 | 4733242 | 40000000.0 | 4505202.0 | 0.0 | 165000.0 | 4766229.0 | 46.7 | 16.90 | 17.10 |
| **9881** | 2 | 2294072 | 40000000.0 | 1115000.0 | 0.0 | 849587.0 | 1925403.0 | 81.7 | 0.00 | 98.80 |
| **5995** | 6 | 22629564 | 180000000.0 | 20209542.0 | 0.0 | 3927461.0 | 22643270.0 | 40.1 | 11.90 | 58.00 |
| **12277** | 2 | 8153311 | 80000000.0 | 3644008.0 | 0.0 | 946400.0 | 7961240.0 | 0.0 | 1.24 | 97.00 |
| **12644** | 2 | 8479101 | 70000000.0 | 8528016.0 | 0.0 | 0.0 | 8464004.0 | 2 | 21.70 | 45.50 |

6198 rows × 22 columns

In [26]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(pred_xgb,y_cv)
```

Out[26]:
```
array([[1528,   81],
       [ 152, 1638]], dtype=int64)
```