1.24 MB

In [3]:

```
import pandas as pd
import sklearn
import numpy as np
import matplotlib.pyplot as plt
import os
import warnings
import seaborn as sns
from sklearn.preprocessing import OneHotEncoc
from sklearn.datasets import make_blobs
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardSca
from sklearn.svm import LinearSVC
from sklearn.metrics import roc_auc_score
from sklearn.linear_model import LogisticRegr
from sklearn.metrics import roc_auc_score
from sklearn.calibration import CalibratedCla
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClas
from sklearn.metrics import accuracy_score
from sklearn.linear_model import SGDClassifie
import plotly.offline as py
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode
from sklearn.model_selection import train_tes
init_notebook_mode(connected=True)
import cufflinks as cf
cf.go_offline()
import pickle
import gc
import lightgbm as lgb
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [4]:

```
house_loan=pd.read_csv('loan_data.csv')
house_loan.describe()
```

Out[4]:

	SK_ID_CURR	TARGET	CNT_CHILDREN
count	307511.000000	307511.000000	307511.000000
mean	278180.518577	0.080729	0.417052
std	102790.175348	0.272419	0.722121
min	100002.000000	0.000000	0.000000
25%	189145.500000	0.000000	0.000000
50%	278202.000000	0.000000	0.000000
75%	367142.500000	0.000000	1.000000
max	456255.000000	1.000000	19.000000

8 rows × 106 columns

In [5]:

```
house_loan.columns
```

Out[5]:

```
Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT
_TYPE', 'CODE_GENDER',
       'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CN
T_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY',
       ...
       'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_1
9', 'FLAG_DOCUMENT_20',
```

```
       'FLAG_DOCUMENT_21', 'AMT_REQ_CREDIT_BU
REAU_HOUR',
       'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_
CREDIT_BUREAU_WEEK',
       'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_
CREDIT_BUREAU_QRT',
       'AMT_REQ_CREDIT_BUREAU_YEAR'],
      dtype='object', length=122)
```

In [6]:

```python
house_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_C
REDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

In [7]:

```python
house_loan.isnull().sum()
```

Out[7]:

```
SK_ID_CURR                        0
TARGET                            0
NAME_CONTRACT_TYPE                0
CODE_GENDER                       0
FLAG_OWN_CAR                      0
                              ...
AMT_REQ_CREDIT_BUREAU_DAY     41519
AMT_REQ_CREDIT_BUREAU_WEEK    41519
AMT_REQ_CREDIT_BUREAU_MON     41519
AMT_REQ_CREDIT_BUREAU_QRT     41519
AMT_REQ_CREDIT_BUREAU_YEAR    41519
Length: 122, dtype: int64
```

In [8]:

```python
house_loan.head()
```

Out[8]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	COI
0	100002	1	Cash loans	
1	100003	0	Cash loans	
2	100004	0	Revolving loans	
3	100006	0	Cash loans	
4	100007	0	Cash loans	

5 rows × 122 columns

In [9]:

```python
defaulters=(house_loan.TARGET==1).sum()
payers=(house_loan.TARGET==0).sum()
print((defaulters/payers)*100)
```

```
8.781828601345662
```

In [10]:

```python
without_id=[column for column in house_loan.c

#check for duplicate values
na=house_loan[house_loan.duplicated(subset=wi
print("Duplicates are: ",na.shape[0])
```

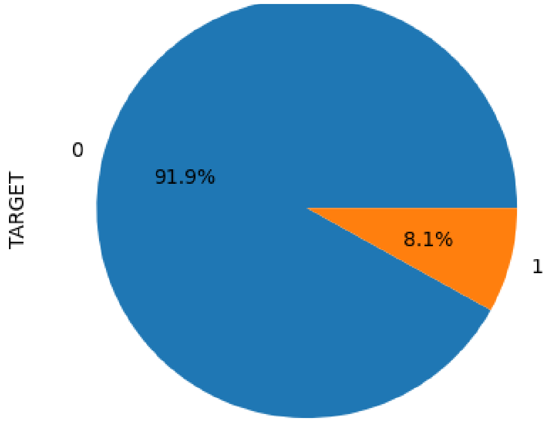
```
Duplicates are:  0
```

In [11]:

```python
house_loan.TARGET.value_counts().plot(kind='p
```

Out[11]:

```
<AxesSubplot: ylabel='TARGET'>
```



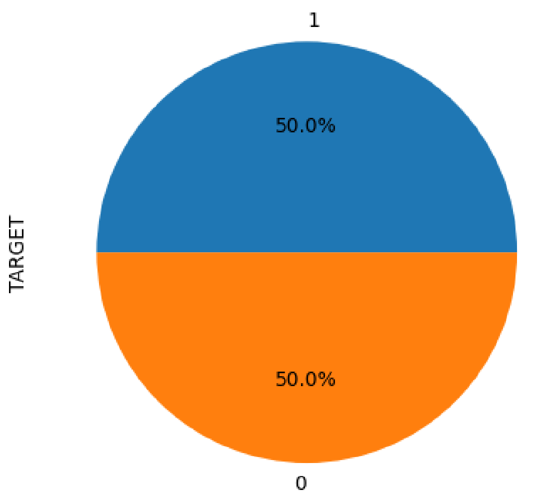
In [12]:

```
import matplotlib as plt
```

In [13]:

```
shuffled_data=house_loan.sample(frac=1,random
unpaid_home_loan=shuffled_data.loc[shuffled_d
paid_home_loan=shuffled_data.loc[shuffled_dat
normalised_home_loan=pd.concat([unpaid_home_l
normalised_home_loan.TARGET.value_counts().pl
```

Out[13]: <AxesSubplot: ylabel='TARGET'>



In [14]:

```
import tensorflow as tf
```

In [15]:

```
normalised_home_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 49650 entries, 207339 to 121862
Columns: 122 entries, SK_ID_CURR to AMT_REQ_C
REDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 46.6+ MB
```

In [16]:

```
normalised_home_loan.head
```

Out[16]:

```
<bound method NDFrame.head of          SK_ID_C
URR  TARGET NAME_CONTRACT_TYPE CODE_GENDER FL
AG_OWN_CAR  \
207339      340318       1         Cash loans
F             N
```

```
8756        110186       1         Cash loans
M            Y
230344      366811       1         Cash loans
F            N
178329      306645       1         Cash loans
M            Y
55586       164407       1         Cash loans
M            N
...            ...     ...                ...
...           ...
130947      251878       0         Cash loans
F            Y
40467       146875       0         Cash loans
F            N
187004      316791       0         Cash loans
M            N
131755      252811       0         Cash loans
F            N
121862      241287       0         Cash loans
M            N

       FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INC
OME_TOTAL  AMT_CREDIT  \
207339               N             0
112500.0    405000.0
8756                 N             0
135000.0    544491.0
230344               Y             0
112500.0    225000.0
178329               Y             0
157500.0    595273.5
55586                N             0
157500.0    521451.0
...                ...           ...
...          ...
130947               Y             0
135000.0    770913.0
40467                N             2
360000.0    260640.0
187004               Y             1
180000.0    688500.0
131755               Y             2
202500.0    312840.0
121862               N             0
58500.0    254700.0

        AMT_ANNUITY  ...  FLAG_DOCUMENT_18 FL
AG_DOCUMENT_19 FLAG_DOCUMENT_20  \
207339      21969.0  ...                 0
0               0
8756        17563.5  ...                 0
0               0
230344      17905.5  ...                 0
0               0
178329      29083.5  ...                 0
0               0
55586       35406.0  ...                 0
0               0
...             ...  ...               ...
...             ...
130947      24997.5  ...                 0
0               0
40467       29475.0  ...                 0
0               0
187004      22752.0  ...                 0
0               0
131755      18090.0  ...                 0
0               0
121862      13446.0  ...                 0
0               0

       FLAG_DOCUMENT_21 AMT_REQ_CREDIT_BUREAU
_HOUR AMT_REQ_CREDIT_BUREAU_DAY  \
207339                  0
```

```
0.0                         0.0
8756                  0
0.0                         0.0
230344                0
NaN                         NaN
178329                0
NaN                         NaN
55586                 0
0.0                         0.0
...                 ...
...                         ...
130947                0
0.0                         0.0
40467                 0
0.0                         0.0
187004                0
0.0                         0.0
131755                0
0.0                         0.0
121862                0
0.0                         0.0

        AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_C
REDIT_BUREAU_MON  \
207339                         0.0
0.0
8756                           0.0
0.0
230344                         NaN
NaN
178329                         NaN
NaN
55586                          0.0
0.0
...                            ...
...
130947                         0.0
1.0
40467                          0.0
0.0
187004                         0.0
0.0
131755                         0.0
0.0
121862                         0.0
0.0

        AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CR
EDIT_BUREAU_YEAR
207339                        0.0
3.0
8756                          0.0
0.0
230344                        NaN
NaN
178329                        NaN
NaN
55586                         0.0
1.0
...                           ...
...
130947                        1.0
1.0
40467                         0.0
0.0
187004                        0.0
0.0
131755                        1.0
3.0
121862                        0.0
0.0

[49650 rows x 122 columns]>
```

In [17]:

```
normalised_home_loan.dropna(axis=0)
normalised_home_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 49650 entries, 207339 to 121862
Columns: 122 entries, SK_ID_CURR to AMT_REQ_C
REDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 46.6+ MB
```

In [18]:

```
normalised_home_loan.isnull().sum()
```

Out[18]:

```
SK_ID_CURR                      0
TARGET                          0
NAME_CONTRACT_TYPE              0
CODE_GENDER                     0
FLAG_OWN_CAR                    0
                             ...
AMT_REQ_CREDIT_BUREAU_DAY    7648
AMT_REQ_CREDIT_BUREAU_WEEK   7648
AMT_REQ_CREDIT_BUREAU_MON    7648
AMT_REQ_CREDIT_BUREAU_QRT    7648
AMT_REQ_CREDIT_BUREAU_YEAR   7648
Length: 122, dtype: int64
```

In [19]:

```
#print(normalised_home_loan.apply())
```

In [20]:

```
print(pd.unique(normalised_home_loan.AMT_REQ_
print(pd.unique(normalised_home_loan.AMT_REQ_
print(pd.unique(normalised_home_loan.AMT_REQ_
print(pd.unique(normalised_home_loan.AMT_REQ_
print(pd.unique(normalised_home_loan.AMT_REQ_
```

```
[ 0. nan  1.  2.  4.  3.  9.]
[ 0. nan  1.  2.  4.  3.  5.  6.]
[ 0. nan  1.  3.  5.  9.  2.  6.  8.  4. 11.
12.  7. 13. 10. 17. 15. 14.
 16. 18. 27.]
[ 0. nan  2.  3.  1.  4.  5.  6. 19.  7.]
[ 3.  0. nan  1.  5.  4.  2.  6.  7.  8.  9.
10. 14. 13. 12. 11. 22. 16.
 23. 17.]
```

In [21]:

```
normalised_home_loan.dropna(axis=0)
```

Out[21]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYP
279124	423360	1	Cash loan
216116	350411	1	Cash loan
133687	255050	1	Cash loan
4159	104863	1	Cash loan
208602	341779	1	Cash loan
...	...	...	.
108677	226053	0	Cash loan
258603	399273	0	Revolving loan
51880	160079	0	Cash loan
282820	427561	0	Cash loan
207101	340051	0	Revolving loan

1230 rows × 122 columns

In [22]:

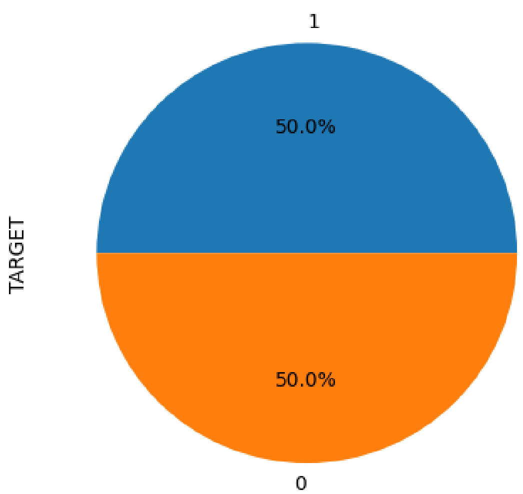
```
print(normalised_home_loan.info())
print(normalised_home_loan.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 49650 entries, 207339 to 121862
Columns: 122 entries, SK_ID_CURR to AMT_REQ_C
REDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 46.6+ MB
None
SK_ID_CURR                       0
TARGET                           0
NAME_CONTRACT_TYPE               0
CODE_GENDER                      0
FLAG_OWN_CAR                     0
                              ...
AMT_REQ_CREDIT_BUREAU_DAY     7648
AMT_REQ_CREDIT_BUREAU_WEEK    7648
AMT_REQ_CREDIT_BUREAU_MON     7648
AMT_REQ_CREDIT_BUREAU_QRT     7648
AMT_REQ_CREDIT_BUREAU_YEAR    7648
Length: 122, dtype: int64
```

In [23]:

```
normalised_home_loan.TARGET.value_counts().pl
```

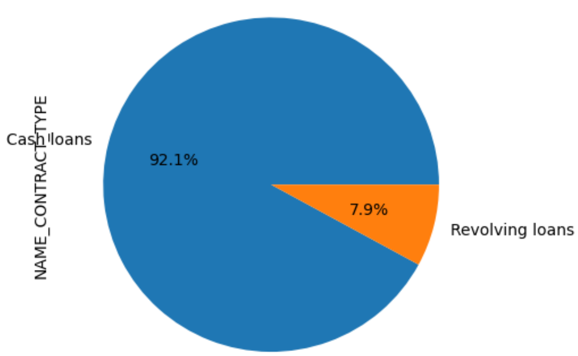
Out[23]: <AxesSubplot: ylabel='TARGET'>



In [24]:

```
normalised_home_loan.NAME_CONTRACT_TYPE.value
#high amount of cash loans
```

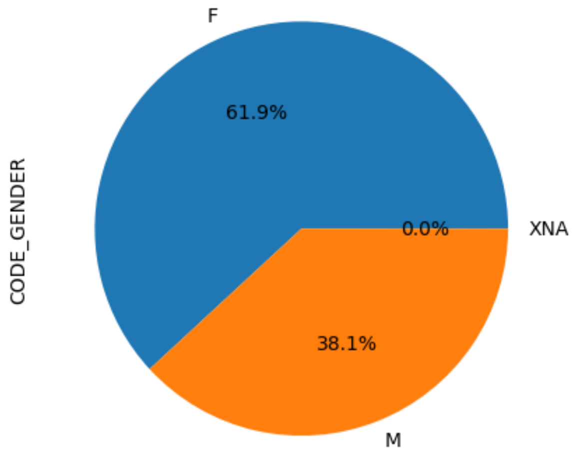
Out[24]: <AxesSubplot: ylabel='NAME_CONTRACT_TYPE'>



In [25]:

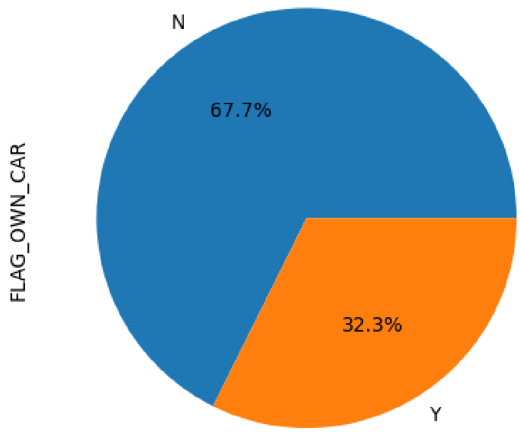
```
normalised_home_loan.CODE_GENDER.value_counts
#roughly equal amount
```

Out[25]: <AxesSubplot: ylabel='CODE_GENDER'>



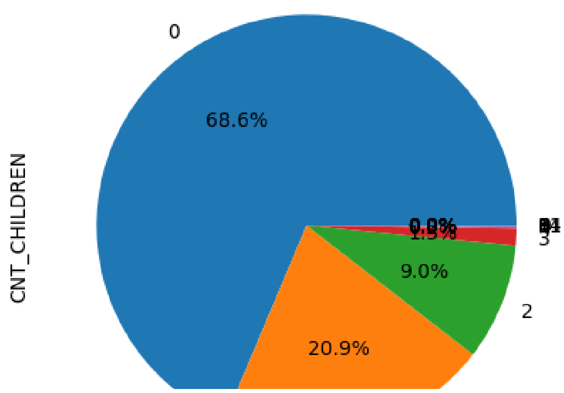
In [26]:
```
normalised_home_loan.FLAG_OWN_CAR.value_count
```

Out[26]: <AxesSubplot: ylabel='FLAG_OWN_CAR'>



In [27]:
```
normalised_home_loan.CNT_CHILDREN.value_count
```

Out[27]: <AxesSubplot: ylabel='CNT_CHILDREN'>



In [28]:

```
#!pip install chart_studio


cf.set_config_file(theme='polar')

normalised_home_loan[normalised_home_loan['AM
   xTitle = 'Total Income', yTitle ='Count of
            title='Distribution of AMT_INCOM
```

In [29]:

```
(normalised_home_loan[normalised_home_loan['A
```

Out[29]:

```
0    64.864865
1    35.135135
Name: TARGET, dtype: float64
```

In [30]:

```
#print((normalised_home_loan[normalised_home_
print((normalised_home_loan[normalised_home_l
print((normalised_home_loan[normalised_home_l
#as number of children is increasing lone def
```

```
1    57.047872
0    42.952128
Name: TARGET, dtype: float64
1    81.818182
0    18.181818
Name: TARGET, dtype: float64
```

In [31]:

```
print((normalised_home_loan[normalised_home_l
print((normalised_home_loan[normalised_home_l

#people with own cars are slighlty more likel
```

```
1    51.350064
0    48.649936
Name: TARGET, dtype: float64
0    52.823962
1    47.176038
Name: TARGET, dtype: float64
```

In [32]:

```
print((normalised_home_loan[normalised_home_l
print((normalised_home_loan[normalised_home_l

#men more likely to default in payment of loa
```

```
1    56.280372
0    43.719628
Name: TARGET, dtype: float64
0    53.867691
1    46.132309
Name: TARGET, dtype: float64
```

In [33]:

```
print((normalised_home_loan[normalised_home_l
print((normalised_home_loan[normalised_home_l

#cash loans have a higher percent of default€
```

```
1    50.802923
0    49.197077
Name: TARGET, dtype: float64
0    59.309995
1    40.690005
Name: TARGET, dtype: float64
```

In [34]:

```
normalised_home_loan=normalised_home_loan.sam
```

In [35]:

```
from sklearn.preprocessing import OrdinalEnco

ordenc=OrdinalEncoder()
normalised_home_loan['NAME_CONTRACT_TYPE_CODE
print(normalised_home_loan[['NAME_CONTRACT_TY
print(normalised_home_loan['NAME_CONTRACT_TYP
```

```
       NAME_CONTRACT_TYPE  NAME_CONTRACT_TYPE
_CODE
302218         Cash loans
0.0
167526         Cash loans
0.0
159305         Cash loans
0.0
275427         Cash loans
0.0
8837           Cash loans
0.0
192094         Cash loans
0.0
235115    Revolving loans
1.0
79051          Cash loans
0.0
123267    Revolving loans
1.0
5517           Cash loans
0.0
128624         Cash loans
0.0
187583         Cash loans
0.0
143193         Cash loans
0.0
288269         Cash loans
0.0
44320          Cash loans
0.0
256898         Cash loans
0.0
118237         Cash loans
0.0
5980      Revolving loans
1.0
96475          Cash loans
0.0
249976         Cash loans
0.0
0.0    45708
1.0     3942
Name: NAME_CONTRACT_TYPE_CODE, dtype: int64
```

In [36]:

```
normalised_home_loan['CODE_GENDER_CODE']=orde
print(normalised_home_loan[['CODE_GENDER','CO
print(normalised_home_loan['CODE_GENDER_CODE'
```

```
       CODE_GENDER  CODE_GENDER_CODE
302218           M               1.0
167526           F               0.0
159305           M               1.0
275427           F               0.0
8837             M               1.0
192094           M               1.0
235115           F               0.0
79051            F               0.0
123267           M               1.0
5517             F               0.0
```

```
128624            M              1.0
187583            F              0.0
143193            M              1.0
288269            F              0.0
44320             F              0.0
256898            F              0.0
118237            F              0.0
5980              M              1.0
96475             F              0.0
249976            F              0.0
0.0    30716
1.0    18932
2.0        2
Name: CODE_GENDER_CODE, dtype: int64
```

In [37]:

```
#2 other values in code_gender
normalised_home_loan.loc[normalised_home_loar
```

Out[37]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPI
83382	196708	0	Revolving loan
189640	319880	0	Revolving loan

2 rows × 124 columns

In [38]:

```
normalised_home_loan['FLAG_OWN_CAR_CODE']=orc
print(normalised_home_loan[['FLAG_OWN_CAR','F
print(normalised_home_loan['FLAG_OWN_CAR_CODE
```

```
       FLAG_OWN_CAR  FLAG_OWN_CAR_CODE
302218            N                0.0
167526            N                0.0
159305            N                0.0
275427            N                0.0
8837              N                0.0
192094            N                0.0
235115            N                0.0
79051             N                0.0
123267            N                0.0
5517              N                0.0
128624            N                0.0
187583            N                0.0
143193            N                0.0
288269            Y                1.0
44320             Y                1.0
256898            N                0.0
118237            N                0.0
5980              Y                1.0
96475             N                0.0
249976            N                0.0
0.0    33591
1.0    16059
Name: FLAG_OWN_CAR_CODE, dtype: int64
```

In [39]:

```
normalised_home_loan['CNT_CHILDREN_CODE']=orc
print(normalised_home_loan[['CNT_CHILDREN_COD
print(normalised_home_loan['CNT_CHILDREN_CODE
```

```
        CNT_CHILDREN_CODE  CNT_CHILDREN
302218                0.0             0
167526                0.0             0
159305                2.0             2
275427                0.0             0
8837                  0.0             0
192094                0.0             0
235115                0.0             0
79051                 0.0             0
123267                1.0             1
5517                  0.0             0
128624                0.0             0
```

```
187583                1.0            1
143193                0.0            0
288269                0.0            0
44320                 0.0            0
256898                0.0            0
118237                2.0            2
5980                  0.0            0
96475                 0.0            0
249976                0.0            0
0.0     34073
1.0     10381
2.0      4444
3.0       642
4.0        89
5.0        10
6.0         6
8.0         2
9.0         1
10.0        1
7.0         1
Name: CNT_CHILDREN_CODE, dtype: int64
```

In [40]:

```
normalised_home_loan=normalised_home_loan.sam
```

In [41]:

```
normalised_home_loan['TARGET'].value_counts()
```

Out[41]:

```
0    24825
1    24825
Name: TARGET, dtype: int64
```

In [42]:

```
y=normalised_home_loan.TARGET
```

In [43]:

```
#y=y.sample(frac=1,random_state=45)
```

In [44]:

```
normalised_home_loan_features=['SK_ID_CURR','
```

In [45]:

```
from sklearn.model_selection import train_tes
```

In [46]:

```
X=normalised_home_loan[normalised_home_loan_f
```

In [47]:

```
#X=X.sample(frac=1,random_state=45)
```

In [48]:

```
blobs_random_seed = 42
centers = [(0,0), (5,5)]
cluster_std = 1
frac_test_split = 0.33
num_features_for_samples = 2
num_samples_total = 49650

# Generate data
inputs, targets = make_blobs(n_samples = num_

X_train,X_test,y_train,y_test=train_test_spli
```

In [49]:

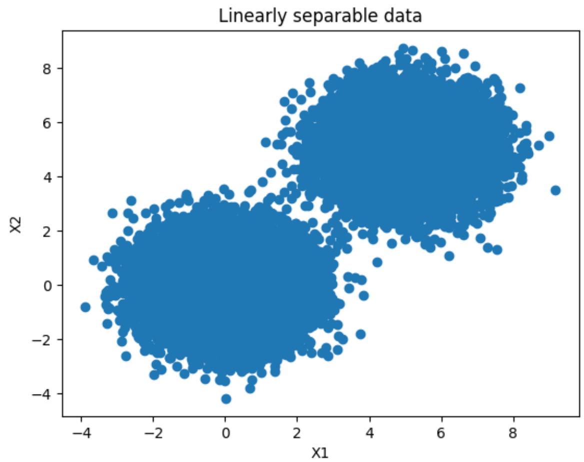
```
print(X_train.shape, X_test.shape, y_train.sh
```

```
(33265, 2) (16385, 2) (33265,) (16385,)
```

In [50]:

```
plt.pyplot.scatter(X_train[:,0], X_train[:,1]
plt.pyplot.title('Linearly separable data')
plt.pyplot.xlabel('X1')
```

```
plt.pyplot.ylabel('X2')
plt.pyplot.show()
```



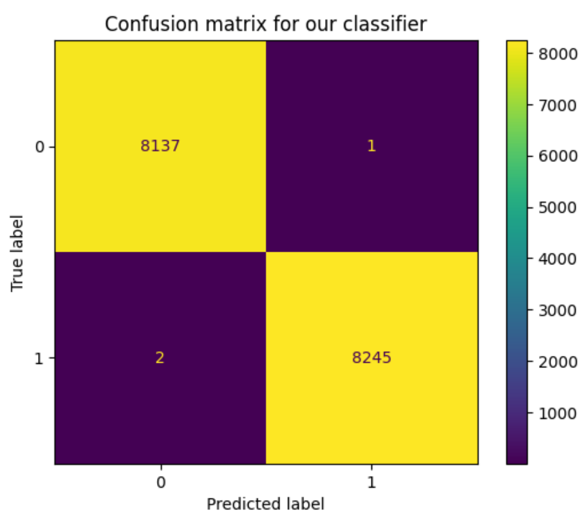
In [51]:
```
from sklearn import svm
from sklearn.metrics import ConfusionMatrixDi
```

In [52]:
```
clf=svm.SVC(kernel='linear')
```

In [53]:
```
clf=clf.fit(X_train,y_train)
```

In [54]:
```
predictions = clf.predict(X_test)

# Generate confusion matrix
matrix = ConfusionMatrixDisplay.from_predicti
plt.pyplot.title('Confusion matrix for our cl
plt.pyplot.show(matrix)
plt.pyplot.show()
```



In [55]:
```
from sklearn.metrics import precision_score,
```

In [56]:
```
print(precision_score(y_test, predictions))
print(recall_score(y_test, predictions))
print(f1_score(y_test,predictions,average=Nor
```

```
0.9998787290807665
0.9997574875712381
[0.99981569 0.9998181 ]
```

In [57]:

```
support_vectors = clf.support_vectors_

# Visualize support vectors
plt.pyplot.scatter(X_train[:,0], X_train[:,1]
plt.pyplot.scatter(support_vectors[:,0], supp
plt.pyplot.title('Linearly separable data wit
plt.pyplot.xlabel('X1')
plt.pyplot.ylabel('X2')
plt.pyplot.show()
```