In [1]:

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

In [2]:

```
Data = pd.read_csv("Tensorflow_Project_Loan_Data.csv")
```

In [3]:

```
Data.head()
```

Out[3]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR
0	100002	1	Cash loans	M	N
1	100003	0	Cash loans	F	N
2	100004	0	Revolving loans	M	Y
3	100006	0	Cash loans	F	N
4	100007	0	Cash loans	M	N

5 rows × 122 columns

In [4]:

```
Data.describe()
```

Out[4]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT
count	236868.000000	236868.000000	236868.000000	2.368680e+05	2.368680e+05
mean	237303.640276	0.081024	0.416734	1.688321e+05	5.990305e+05
std	79217.796677	0.272873	0.722105	2.632733e+05	4.021758e+05
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04
25%	168636.750000	0.000000	0.000000	1.125000e+05	2.700000e+05
50%	237331.500000	0.000000	0.000000	1.440000e+05	5.135310e+05
75%	305866.250000	0.000000	1.000000	2.025000e+05	8.086500e+05
max	374360.000000	1.000000	19.000000	1.170000e+08	4.050000e+06

8 rows × 106 columns

In [5]:

```
Data.columns
```

Out[5]:

```
Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
       'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTA
L',
       'AMT_CREDIT', 'AMT_ANNUITY',
       ...
       'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20',
       'FLAG_DOCUMENT_21', 'AMT_REQ_CREDIT_BUREAU_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]:

```
Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236868 entries, 0 to 236867
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(85), int64(21), object(16)
memory usage: 220.5+ MB
```

In [7]:

```
# Check the null values.
Data.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     32054
AMT_REQ_CREDIT_BUREAU_WEEK    32054
AMT_REQ_CREDIT_BUREAU_MON     32054
AMT_REQ_CREDIT_BUREAU_QRT     32054
AMT_REQ_CREDIT_BUREAU_YEAR    32054
Length: 122, dtype: int64
```

In [8]:

```
#
defaulters = (Data.TARGET==1).sum()
payers = (Data.TARGET==0).sum()
print((defaulters/payers)*100)
```

```
8.816773553354528
```

In [9]:

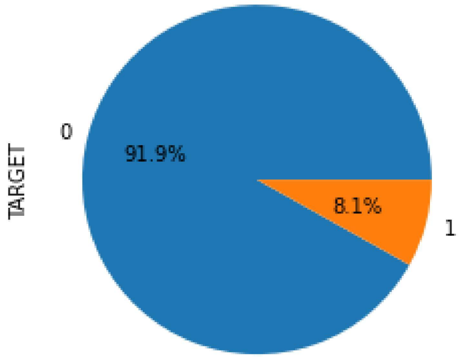
```
import tensorflow as tf
from tensorflow import keras
```

In [10]:

```
Data.TARGET.value_counts().plot(kind='pie',autopct = '%1.1f%%')
```

Out[10]:

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



In [11]:

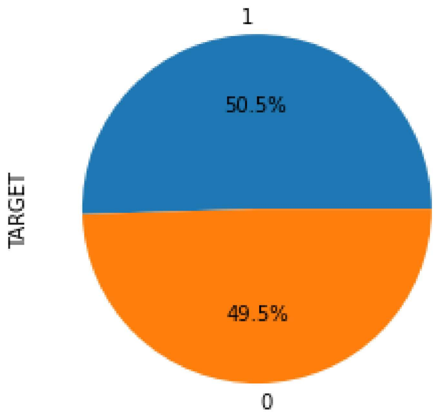
```
shuffled_data = Data.sample(frac=1, random_state=3)
unpaid_home_loan = shuffled_data.loc[shuffled_data['TARGET']==1]
paid_home_loan = shuffled_data.loc[shuffled_data['TARGET']==0].sample(n=18825, random_s
tate=69)
normalised_home_loan = pd.concat([unpaid_home_loan,paid_home_loan])
```

In [12]:

```
normalised_home_loan.TARGET.value_counts().plot(kind='pie',autopct = '%1.1f%%')
```

Out[12]:

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



In [13]:

```
normalised_home_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38017 entries, 115045 to 135625
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(85), int64(21), object(16)
memory usage: 35.7+ MB
```

In [14]:

```
normalised_home_loan.head
```

Out[14]:

```
<bound method NDFrame.head of         SK_ID_CURR  TARGET NAME_CONTRACT_TYP
E CODE_GENDER FLAG_OWN_CAR  \
115045      233398       1         Cash loans           F            N
82905       196155       1         Cash loans           F            N
117030      235708       1         Cash loans           F            Y
93435       208504       1         Cash loans           F            N
141971      264612       1         Cash loans           M            N
...            ...     ...                ...         ...          ...
155659      280439       0         Cash loans           M            N
132613      253810       0         Cash loans           M            Y
78648       191170       0    Revolving loans           F            Y
145980      269265       0         Cash loans           M            Y
135625      257306       0         Cash loans           F            N

       FLAG_OWN_REALTY  CNT_CHILDREN  AMT_INCOME_TOTAL  AMT_CREDIT  \
115045               Y             0          135000.0    417024.0
82905                Y             0          112500.0    508495.5
117030               Y             0          292500.0   1252278.0
93435                Y             0           90000.0    152820.0
141971               N             0          270000.0    450000.0
...                ...           ...               ...         ...
155659               Y             1          126000.0    270000.0
132613               Y             0          162000.0    197820.0
78648                N             0           45000.0    202500.0
145980               Y             0          360000.0   1157670.0
135625               Y             0          315000.0    835380.0

        AMT_ANNUITY  ...  FLAG_DOCUMENT_18 FLAG_DOCUMENT_19 FLAG_DOCUMENT_
20  \
115045      28341.0  ...               0.0              0.0
0.0
82905       21541.5  ...               0.0              0.0
0.0
117030      36747.0  ...               0.0              0.0
0.0
93435        9895.5  ...               0.0              0.0
0.0
141971      22018.5  ...               0.0              0.0
0.0
...             ...  ...               ...              ...
...
155659      16443.0  ...               0.0              0.0
0.0
132613      13896.0  ...               0.0              0.0
0.0
78648       10125.0  ...               0.0              0.0
0.0
145980     112909.5  ...               0.0              0.0
0.0
135625      40320.0  ...               0.0              0.0
0.0

       FLAG_DOCUMENT_21 AMT_REQ_CREDIT_BUREAU_HOUR AMT_REQ_CREDIT_BUREAU_D
AY  \
115045              0.0                        0.0
0.0
82905               0.0                        NaN                       N
aN
117030              0.0                        0.0
```

```
0.0
93435                0.0                        0.0
0.0
141971               0.0                        0.0
0.0
...                  ...                        ...
...
155659               0.0                        0.0
0.0
132613               0.0                        0.0
0.0
78648                0.0                        0.0
0.0
145980               0.0                        0.0
0.0
135625               0.0                        0.0
0.0

        AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
115045                         0.0                        0.0
82905                          NaN                        NaN
117030                         0.0                        0.0
93435                          0.0                        0.0
141971                         0.0                        0.0
...                            ...                        ...
155659                         0.0                        0.0
132613                         1.0                        0.0
78648                          0.0                        0.0
145980                         0.0                        0.0
135625                         0.0                        0.0

        AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
115045                        1.0                         1.0
82905                         NaN                         NaN
117030                        0.0                         6.0
93435                         0.0                         6.0
141971                        2.0                         4.0
...                           ...                         ...
155659                        0.0                         0.0
132613                        0.0                         2.0
78648                         0.0                         1.0
145980                        1.0                         3.0
135625                        0.0                         7.0

[38017 rows x 122 columns]>
```

In [15]:

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38017 entries, 115045 to 135625
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(85), int64(21), object(16)
memory usage: 35.7+ MB
```

In [16]:

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

Out[16]:

```
SK_ID_CURR                     0
TARGET                         0
NAME_CONTRACT_TYPE             0
CODE_GENDER                    0
FLAG_OWN_CAR                   0
                            ... 
AMT_REQ_CREDIT_BUREAU_DAY   5820
AMT_REQ_CREDIT_BUREAU_WEEK  5820
AMT_REQ_CREDIT_BUREAU_MON   5820
AMT_REQ_CREDIT_BUREAU_QRT   5820
AMT_REQ_CREDIT_BUREAU_YEAR  5820
Length: 122, dtype: int64
```

In [17]:

```
print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_DAY))
```

```
[ 0. nan  1.  2.  4.  3.  8.]
```

In [18]:

```
print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_WEEK))
print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_MON))
print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_QRT))
print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_YEAR))
```

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

In [19]:

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

Out[19]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR
196642	328006	1	Cash loans	F	Y
83640	197008	1	Cash loans	F	Y
235517	372801	1	Cash loans	M	Y
129637	250360	1	Cash loans	M	Y
99061	215014	1	Cash loans	F	Y
...	...	...	...	...	...
182309	311302	0	Cash loans	M	Y
152919	277238	0	Cash loans	M	Y
224747	360313	0	Cash loans	F	Y
62985	173059	0	Cash loans	M	Y
195626	326836	0	Cash loans	M	Y

908 rows × 122 columns

In [20]:

```
print(normalised_home_loan.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38017 entries, 115045 to 135625
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(85), int64(21), object(16)
memory usage: 35.7+ MB
None
```

In [21]:

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

```
SK_ID_CURR                       0
TARGET                           0
NAME_CONTRACT_TYPE               0
CODE_GENDER                      0
FLAG_OWN_CAR                     0
                              ...
AMT_REQ_CREDIT_BUREAU_DAY     5820
AMT_REQ_CREDIT_BUREAU_WEEK    5820
AMT_REQ_CREDIT_BUREAU_MON     5820
AMT_REQ_CREDIT_BUREAU_QRT     5820
AMT_REQ_CREDIT_BUREAU_YEAR    5820
Length: 122, dtype: int64
```

In [22]:

```
(normalised_home_loan[normalised_home_loan['AMT_INCOME_TOTAL']>1000000]['TARGET'].value
_counts())/len(normalised_home_loan[normalised_home_loan['AMT_INCOME_TOTAL']>1000000])*
100
```

Out[22]:

```
0    62.962963
1    37.037037
Name: TARGET, dtype: float64
```

In [25]:

```
print((normalised_home_loan[normalised_home_loan['CNT_CHILDREN']>2]['TARGET'].value_cou
nts())/len(normalised_home_loan[normalised_home_loan['CNT_CHILDREN'] > 2])*100)
```

```
1    54.700855
0    45.299145
Name: TARGET, dtype: float64
```

In [26]:

```
print((normalised_home_loan[normalised_home_loan['CNT_CHILDREN']>5]['TARGET'].value_cou
nts())/len(normalised_home_loan[normalised_home_loan['CNT_CHILDREN'] > 5])*100)
```

```
1    80.0
0    20.0
Name: TARGET, dtype: float64
```

In [27]:

```
print((normalised_home_loan[normalised_home_loan['FLAG_OWN_CAR']=='N']['TARGET'].value_
counts())/len(normalised_home_loan[normalised_home_loan['FLAG_OWN_CAR']=='N'])*100)
```

```
1    51.92188
0    48.07812
Name: TARGET, dtype: float64
```

In [28]:

```
print((normalised_home_loan[normalised_home_loan['FLAG_OWN_CAR']=='Y']['TARGET'].value_
counts())/len(normalised_home_loan[normalised_home_loan['FLAG_OWN_CAR']=='Y'])*100)
```

```
0    52.521725
1    47.478275
Name: TARGET, dtype: float64
```

In [29]:

```
print((normalised_home_loan[normalised_home_loan['CODE_GENDER']=='M']['TARGET'].value_c
ounts())/len(normalised_home_loan[normalised_home_loan['CODE_GENDER']=='M'])*100)
```

```
1    56.663241
0    43.336759
Name: TARGET, dtype: float64
```

In [30]:

```
print((normalised_home_loan[normalised_home_loan['CODE_GENDER']=='F']['TARGET'].value_c
ounts())/len(normalised_home_loan[normalised_home_loan['CODE_GENDER']=='F'])*100)
```

```
0    53.368628
1    46.631372
Name: TARGET, dtype: float64
```

In [33]:

```
print((normalised_home_loan[normalised_home_loan['NAME_CONTRACT_TYPE']=='Cash loans'][
'TARGET'].value_counts())/len(normalised_home_loan[normalised_home_loan['NAME_CONTRACT_
TYPE']=='Cash loans'])*100)
```

```
1    51.360788
0    48.639212
Name: TARGET, dtype: float64
```

In [34]:

```
print((normalised_home_loan[normalised_home_loan['NAME_CONTRACT_TYPE']=='Revolving loan
s']['TARGET'].value_counts())/len(normalised_home_loan[normalised_home_loan['NAME_CONTR
ACT_TYPE']=='Revolving loans'])*100)
```

```
0    59.499024
1    40.500976
Name: TARGET, dtype: float64
```

In [35]:

```
normalised_home_loan=normalised_home_loan.sample(frac=1, random_state=5)
```

In [36]:

```
from sklearn.preprocessing import OrdinalEncoder
```

In [38]:

```
ord=OrdinalEncoder()
normalised_home_loan['NAME_CONTRACT_TYPE_CODE']=ord.fit_transform(normalised_home_loan
[['NAME_CONTRACT_TYPE']])
print(normalised_home_loan[['NAME_CONTRACT_TYPE', 'NAME_CONTRACT_TYPE_CODE']].head(20))
```

```
       NAME_CONTRACT_TYPE  NAME_CONTRACT_TYPE_CODE
119660         Cash loans                      0.0
35406          Cash loans                      0.0
213689         Cash loans                      0.0
230729         Cash loans                      0.0
94851          Cash loans                      0.0
12332          Cash loans                      0.0
192312         Cash loans                      0.0
183847    Revolving loans                      1.0
28916          Cash loans                      0.0
10786          Cash loans                      0.0
167695         Cash loans                      0.0
123171         Cash loans                      0.0
51965          Cash loans                      0.0
12998          Cash loans                      0.0
20192          Cash loans                      0.0
210528         Cash loans                      0.0
102497         Cash loans                      0.0
54460          Cash loans                      0.0
61244          Cash loans                      0.0
20954          Cash loans                      0.0
```

In [39]:

```
print(normalised_home_loan['NAME_CONTRACT_TYPE_CODE'].value_counts())
```

```
0.0    34943
1.0     3074
Name: NAME_CONTRACT_TYPE_CODE, dtype: int64
```

In [40]:

```
normalised_home_loan['CODE_GENDER_CODE']=ord.fit_transform(normalised_home_loan[['CODE_GENDER']])
print(normalised_home_loan[['CODE_GENDER', 'CODE_GENDER_CODE']].head(20))
print(normalised_home_loan['CODE_GENDER_CODE'].value_counts())
```

```
       CODE_GENDER  CODE_GENDER_CODE
119660           F               0.0
35406            F               0.0
213689           F               0.0
230729           M               1.0
94851            M               1.0
12332            F               0.0
192312           F               0.0
183847           M               1.0
28916            F               0.0
10786            M               1.0
167695           F               0.0
123171           F               0.0
51965            M               1.0
12998            F               0.0
20192            F               0.0
210528           F               0.0
102497           F               0.0
54460            M               1.0
61244            F               0.0
20954            F               0.0
0.0    23422
1.0    14595
Name: CODE_GENDER_CODE, dtype: int64
```

In [43]:

```
normalised_home_loan.loc[normalised_home_loan['CODE_GENDER_CODE']==2]
```

Out[43]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_

0 rows × 124 columns

In [45]:

```
normalised_home_loan['FLAG_OWN_CAR_CODE']=ord.fit_transform(normalised_home_loan[['FLAG_OWN_CAR']])
print(normalised_home_loan[['FLAG_OWN_CAR', 'FLAG_OWN_CAR_CODE']].head(20))
print(normalised_home_loan['FLAG_OWN_CAR_CODE'].value_counts())
```

```
       FLAG_OWN_CAR  FLAG_OWN_CAR_CODE
119660            N                0.0
35406             N                0.0
213689            Y                1.0
230729            N                0.0
94851             Y                1.0
12332             N                0.0
192312            N                0.0
183847            Y                1.0
28916             N                0.0
10786             N                0.0
167695            N                0.0
123171            N                0.0
51965             Y                1.0
12998             N                0.0
20192             N                0.0
210528            N                0.0
102497            N                0.0
54460             Y                1.0
61244             N                0.0
20954             N                0.0
0.0    25704
1.0    12313
Name: FLAG_OWN_CAR_CODE, dtype: int64
```

In [47]:

```
normalised_home_loan['CNT_CHILDREN_CODE']=ord.fit_transform(normalised_home_loan[['CNT_
CHILDREN']])
print(normalised_home_loan[['CNT_CHILDREN_CODE', 'CNT_CHILDREN']].head(20))
print(normalised_home_loan['CNT_CHILDREN_CODE'].value_counts())
```

```
        CNT_CHILDREN_CODE  CNT_CHILDREN
119660                0.0             0
35406                 2.0             2
213689                0.0             0
230729                0.0             0
94851                 0.0             0
12332                 2.0             2
192312                0.0             0
183847                2.0             2
28916                 0.0             0
10786                 0.0             0
167695                0.0             0
123171                3.0             3
51965                 3.0             3
12998                 0.0             0
20192                 0.0             0
210528                0.0             0
102497                0.0             0
54460                 0.0             0
61244                 0.0             0
20954                 0.0             0
0.0    26011
1.0     7978
2.0     3443
3.0      501
4.0       66
5.0        8
6.0        6
7.0        2
9.0        1
8.0        1
Name: CNT_CHILDREN_CODE, dtype: int64
```

In [48]:

```
normalised_home_loan= normalised_home_loan.sample(frac=1, random_state=45)
```

In [49]:

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

Out[49]:

```
1    19192
0    18825
Name: TARGET, dtype: int64
```

In [50]:

```
y=normalised_home_loan.TARGET
```

In [51]:

```
normalised_home_loan_features=['SK_ID_CURR', 'NAME_CONTRACT_TYPE_CODE', 'CNT_CHILDREN_CODE', 'FLAG_OWN_CAR_CODE', 'CODE_GENDER_CODE']
```

In [52]:

```
from sklearn.model_selection import train_test_split
```

In [67]:

```
X=normalised_home_loan[normalised_home_loan_features]
```

In [68]:

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

In [69]:

```
from sklearn.datasets import make_blobs
```

In [72]:

```
inputs, targets = make_blobs(n_samples = num_sample_total, centers = centers, n_features = num_features_for_sample, cluster_std = cluster_std)
```

In [73]:

```
x_train,x_test,y_train,y_test=train_test_split(inputs, targets, test_size=0.33, random_state=45)
```

In [74]:

```
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

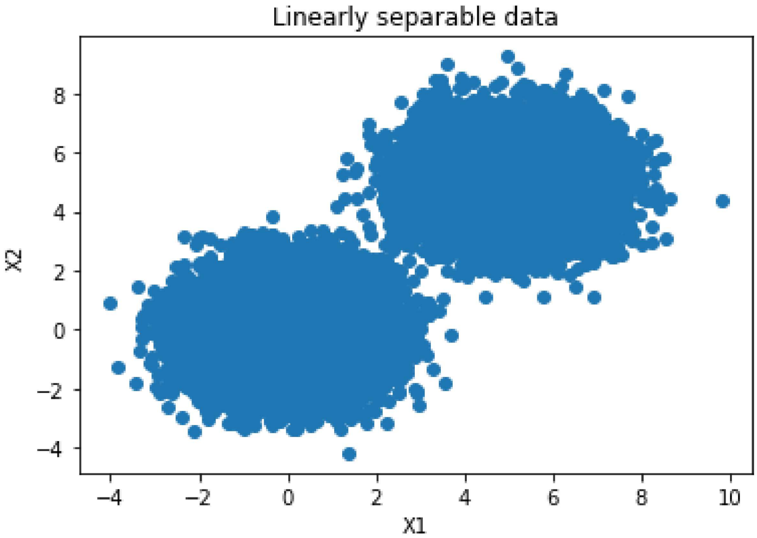
```
(33265, 2) (16385, 2) (33265,) (16385,)
```

In [88]:

```
import matplotlib.pyplot as plt
```

In [89]:

```
plt.scatter(x_train[:,0], x_train[:,1])
plt.title('Linearly separable data')
plt.xlabel('X1')
plt.ylabel('X2')
plt.show()
```



In [90]:

```
from sklearn import svm
from sklearn.metrics import plot_confusion_matrix
```

In [91]:

```
clf = svm.SVC(kernel='linear')
```

In [92]:

```
clf=clf.fit(x_train, y_train)
```

In [94]:

```
predictions = clf.predict(x_test)
```

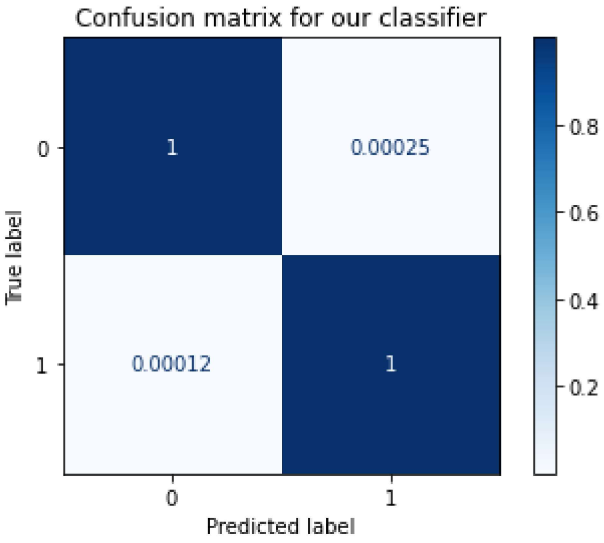
In [97]:

```
matrix = plot_confusion_matrix(clf, x_test, y_test,

cmap = plt.cm.Blues,
normalize='true')
plt.title('Confusion matrix for our classifier')
plt.show(matrix)
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/sklearn/utils/deprecation.py:87: Fu
tureWarning:

Function plot_confusion_matrix is deprecated; Function `plot_confusion_mat
rix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class
methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDispla
y.from_estimator.
```



In [98]:

```
from sklearn.metrics import precision_score, recall_score, f1_score
```

In [99]:

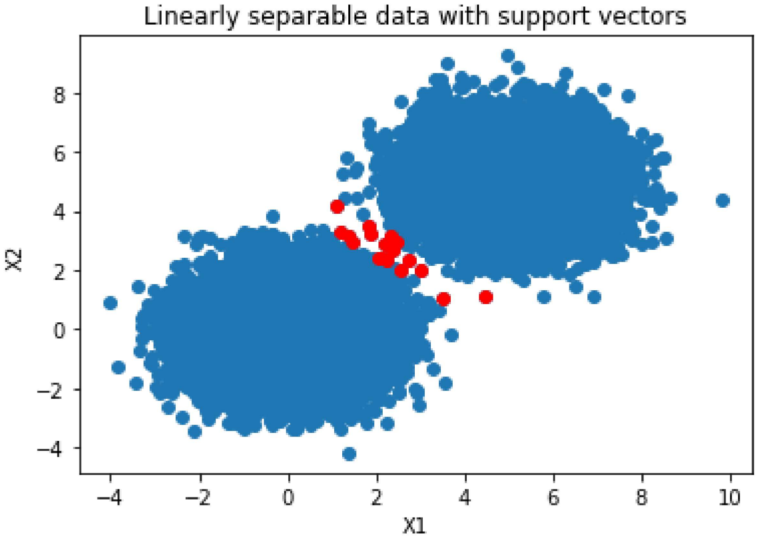
```
print(precision_score(y_test, predictions))
print(recall_score(y_test, predictions))
print(f1_score(y_test, predictions, average=None))
```

```
0.9997568980187188
0.99987843423292
[0.99981614 0.99981766]
```

In [100]:

```
support_vectors = clf.support_vectors_

plt.scatter(x_train[:,0], x_train[:,1])
plt.scatter(support_vectors[:,0], support_vectors[:,1], color = 'red')
plt.title('Linearly separable data with support vectors')
plt.xlabel('X1')
plt.ylabel('X2')
plt.show()
```

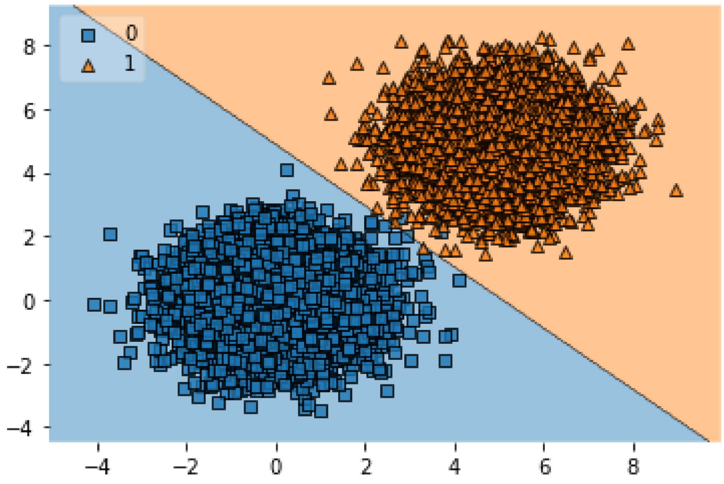


In [101]:

```
from mlxtend.plotting import plot_decision_regions
```

In [102]:

```
plot_decision_regions(x_test, y_test, clf=clf, legend=2)
plt.show()
```



In [ ]:

```

```