

## House Loan Data Analysis

```
In [101]: 1 import numpy as np # Linear algebra
2 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
3
4 # Input data files are available in the read-only "../input/" directory
5 # For example, running this (by clicking run or pressing Shift+Enter) will L
6
7 import os
8 import pandas as pd
9 import sklearn
10 import numpy as np
11 import matplotlib.pyplot as plt
12 import os
13 import warnings
14 import seaborn as sns
15 from sklearn.preprocessing import OneHotEncoder
16 from sklearn.datasets import make_blobs
17 from sklearn.impute import SimpleImputer
18 from sklearn.pipeline import Pipeline
19 from sklearn.compose import ColumnTransformer
20 from sklearn.preprocessing import StandardScaler
21 from sklearn.svm import LinearSVC
22 from sklearn.metrics import roc_auc_score
23 from sklearn.linear_model import LogisticRegression
24 from sklearn.metrics import roc_auc_score
25 from sklearn.calibration import CalibratedClassifierCV
26 from sklearn.metrics import confusion_matrix
27 from sklearn.ensemble import RandomForestClassifier
28 from sklearn.metrics import accuracy_score
29 from sklearn.linear_model import SGDClassifier
30 import plotly.offline as py
31 import plotly.graph_objs as go
32 from plotly.offline import init_notebook_mode, iplot
33 from sklearn.model_selection import train_test_split
34 init_notebook_mode(connected=True)
35 #import cufflinks as cf
36 #cf.go_offline()
37 import pickle
38 import gc
39 #import lightgbm as lgb
40 warnings.filterwarnings('ignore')
41 %matplotlib inline
```

```
In [76]: 1 house_loan=pd.read_csv('loan_data.csv')
        2 house_loan.describe()
```

```
Out[76]:
```

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_A
<b>count</b>	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499
<b>mean</b>	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108
<b>std</b>	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493
<b>min</b>	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615
<b>25%</b>	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524
<b>50%</b>	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903
<b>75%</b>	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596
<b>max</b>	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025

8 rows × 106 columns

```
In [77]: 1 house_loan.columns
```

```
Out[77]: Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
               'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
               '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 [78]: 1 house_loan.info()
```

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

In [79]: 1 house\_loan.isnull().sum()

Out[79]: 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 [80]: 1 house\_loan.head()

Out[80]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN
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 [81]: 1 defaulters=(house\_loan.TARGET==1).sum()  
 2 payers=(house\_loan.TARGET==0).sum()  
 3 print((defaulters/payers)\*100)

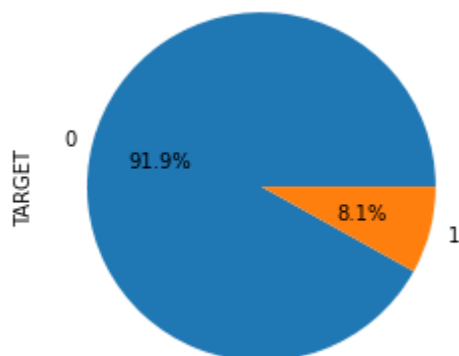
8.781828601345662

In [82]: 1 without\_id=[column for column in house\_loan.columns if column!='SK\_ID\_CURR']  
 2  
 3 #check for duplicate values  
 4 na=house\_loan[house\_loan.duplicated(subset=without\_id,keep=False)]  
 5 print("Duplicates are: ",na.shape[0])

Duplicates are: 0

```
In [83]: 1 house_loan.TARGET.value_counts().plot(kind='pie', autopct='%1.1f%%')
```

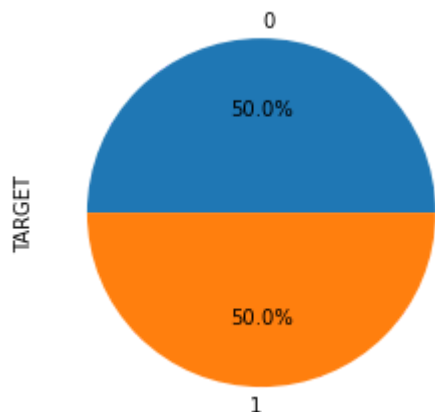
```
Out[83]: <AxesSubplot:ylabel='TARGET'>
```



```
In [84]: 1 import matplotlib as plt
```

```
In [85]: 1 shuffled_data=house_loan.sample(frac=1,random_state=3)
2 unpaid_home_loan=shuffled_data.loc[shuffled_data['TARGET']==1]
3 paid_home_loan=shuffled_data.loc[shuffled_data['TARGET']==0].sample(n=24825,
4 normalised_home_loan=pd.concat([unpaid_home_loan,paid_home_loan])
5 normalised_home_loan.TARGET.value_counts().plot(kind='pie', autopct="%1.1f%%")
```

```
Out[85]: <AxesSubplot:ylabel='TARGET'>
```



In [86]: 1 **import** tensorflow **as** tf

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
<ipython-input-86-64156d691fe5> in <module>  
----> 1 import tensorflow as tf  
  
ModuleNotFoundError: No module named 'tensorflow'
```

In [87]: 1 normalised\_home\_loan.info()

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

In [88]: 1 normalised\_home\_loan.head

```
Out[88]: <bound method NDFrame.head of
E_GENDER FLAG_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_INCOME_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 FLAG_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_CREDIT_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_CREDIT_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 [89]:

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

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

In [90]:

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

```
Out[90]: 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 [91]:

```

1 print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_DAY))
2 print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_WEEK))
3 print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_MON))
4 print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_QRT))
5 print(pd.unique(normalised_home_loan.AMT_REQ_CREDIT_BUREAU_YEAR))

```

```

[ 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.]

```

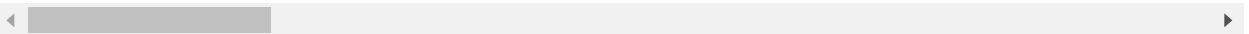
In [92]:

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

Out[92]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG
279124	423360	1	Cash loans	M	Y	
216116	350411	1	Cash loans	M	Y	
133687	255050	1	Cash loans	M	Y	
4159	104863	1	Cash loans	M	Y	
208602	341779	1	Cash loans	F	Y	
...	...	...	...	...	...	...
108677	226053	0	Cash loans	M	Y	
258603	399273	0	Revolving loans	M	Y	
51880	160079	0	Cash loans	M	Y	
282820	427561	0	Cash loans	F	Y	
207101	340051	0	Revolving loans	F	Y	

1230 rows × 122 columns





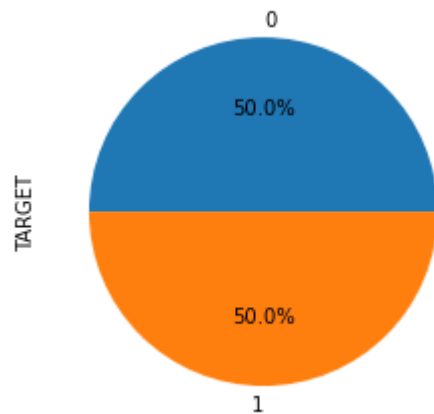
```
In [93]: 1 print(normalised_home_loan.info())
2 print(normalised_home_loan.isnull().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 49650 entries, 207339 to 121862
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_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
```

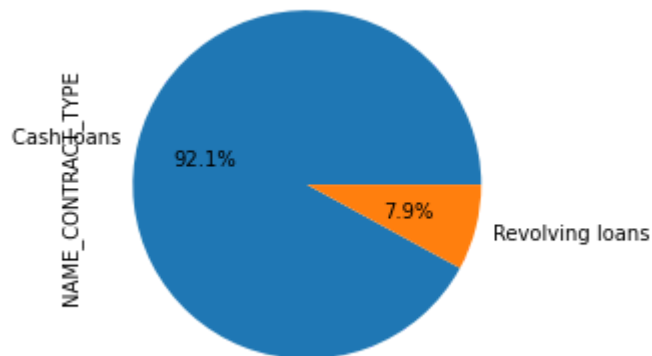
```
In [94]: 1 normalised_home_loan.TARGET.value_counts().plot(kind='pie', autopct="%1.1f%%")
```

```
Out[94]: <AxesSubplot:ylabel='TARGET'>
```



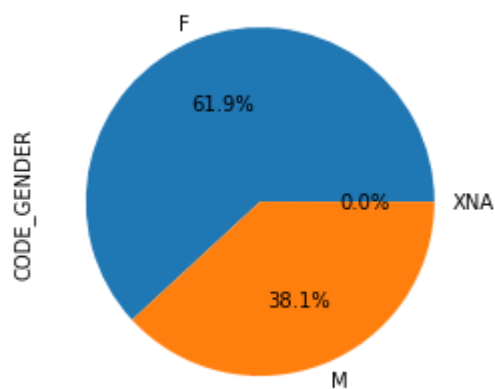
```
In [95]: 1 normalised_home_loan.NAME_CONTRACT_TYPE.value_counts().plot(kind='pie', autopct='%1.1f%%',
2         #high amount of cash loans
```

```
Out[95]: <AxesSubplot:ylabel='NAME_CONTRACT_TYPE'>
```



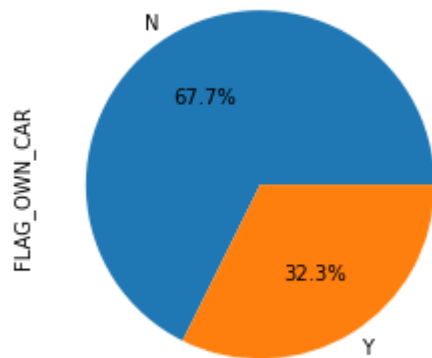
```
In [96]: 1 normalised_home_loan.CODE_GENDER.value_counts().plot(kind='pie', autopct='%1.1f%%',
2         #roughly equal amount
```

```
Out[96]: <AxesSubplot:ylabel='CODE_GENDER'>
```



```
In [97]: 1 normalised_home_loan.FLAG_OWN_CAR.value_counts().plot(kind='pie', autopct="%1
```

```
Out[97]: <AxesSubplot:ylabel='FLAG_OWN_CAR'>
```



```
In [ ]: 1 normalised_home_loan.CNT_CHILDREN.value_counts().plot(kind='pie', autopct="%1
```

```
In [102]: 1 !pip install chart_studio
2
3
4 cf.set_config_file(theme='polar')
5
6 normalised_home_loan[normalised_home_loan['AMT_INCOME_TOTAL'] < 2000000]['AM
7     xTitle = 'Total Income', yTitle = 'Count of applicants',
8     title='Distribution of AMT_INCOME_TOTAL')
9
```

Requirement already satisfied: chart\_studio in c:\users\admin\anaconda3\lib\site-packages (1.1.0)  
 Requirement already satisfied: requests in c:\users\admin\anaconda3\lib\site-packages (from chart\_studio) (2.25.1)  
 Requirement already satisfied: six in c:\users\admin\anaconda3\lib\site-packages (from chart\_studio) (1.15.0)  
 Requirement already satisfied: plotly in c:\users\admin\anaconda3\lib\site-packages (from chart\_studio) (4.14.3)  
 Requirement already satisfied: retrying>=1.3.3 in c:\users\admin\anaconda3\lib\site-packages (from chart\_studio) (1.3.3)  
 Requirement already satisfied: idna<3,>=2.5 in c:\users\admin\anaconda3\lib\site-packages (from requests->chart\_studio) (2.10)  
 Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\admin\anaconda3\lib\site-packages (from requests->chart\_studio) (4.0.0)  
 Requirement already satisfied: certifi>=2017.4.17 in c:\users\admin\anaconda3\lib\site-packages (from requests->chart\_studio) (2020.12.5)  
 Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\admin\anaconda3\lib\site-packages (from requests->chart\_studio) (1.26.4)

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-102-7659de415a57> in <module>
      2
      3
----> 4 cf.set_config_file(theme='polar')
      5
      6 normalised_home_loan[normalised_home_loan['AMT_INCOME_TOTAL'] < 200000
0]['AMT_INCOME_TOTAL'].iplot(kind='histogram', bins=100,
```

**NameError:** name 'cf' is not defined

```
In [103]: 1 (normalised_home_loan[normalised_home_loan['AMT_INCOME_TOTAL'] > 1000000]['TARGET'])
```

```
Out[103]: 0    64.864865
1    35.135135
Name: TARGET, dtype: float64
```

```
In [104]: 1 #print((normalised_home_loan[normalised_home_loan['CNT_CHILDREN']>1]['TARGET']  
2 print((normalised_home_loan[normalised_home_loan['CNT_CHILDREN']>2]['TARGET']  
3 print((normalised_home_loan[normalised_home_loan['CNT_CHILDREN']>5]['TARGET']  
4 #as number of children is increasing lone defaulters are increasing
```

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

```
In [105]: 1 print((normalised_home_loan[normalised_home_loan['CODE_GENDER']=='M']['TARGET']  
2 print((normalised_home_loan[normalised_home_loan['CODE_GENDER']=='F']['TARGET']  
3
```

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

```
In [106]: 1 print((normalised_home_loan[normalised_home_loan['NAME_CONTRACT_TYPE']=='Cas']  
2 print((normalised_home_loan[normalised_home_loan['NAME_CONTRACT_TYPE']=='Rev']  
3  
4 #cash loans have a higher percent of defaulters
```

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

```
In [107]: 1 normalised_home_loan=normalised_home_loan.sample(frac=1,random_state=5)
```

In [108]:

```

1 from sklearn.preprocessing import OrdinalEncoder
2
3 ordenc=OrdinalEncoder()
4 normalised_home_loan['NAME_CONTRACT_TYPE_CODE']=ordenc.fit_transform(normali
5 print(normalised_home_loan[['NAME_CONTRACT_TYPE', 'NAME_CONTRACT_TYPE_CODE']])
6 print(normalised_home_loan['NAME_CONTRACT_TYPE_CODE'].value_counts())

```

	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 [109]: 1 normalised_home_loan['CODE_GENDER_CODE']=ordenc.fit_transform(normalised_hom
2 print(normalised_home_loan[['CODE_GENDER','CODE_GENDER_CODE']].head(20))
3 print(normalised_home_loan['CODE_GENDER_CODE'].value_counts())
```

	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	

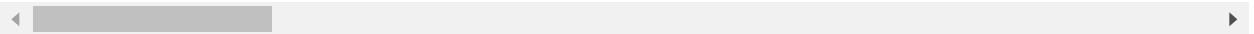
Name: CODE\_GENDER\_CODE, dtype: int64

```
In [110]: 1 #2 other values in code_gender
2 normalised_home_loan.loc[normalised_home_loan['CODE_GENDER_CODE']==2]
```

```
Out[110]:
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG
<b>83382</b>	196708	0	Revolving loans	XNA	N	
<b>189640</b>	319880	0	Revolving loans	XNA	Y	

2 rows × 124 columns



```
In [111]: 1 normalised_home_loan['FLAG_OWN_CAR_CODE']=ordenc.fit_transform(normalised_ho
2 print(normalised_home_loan[['FLAG_OWN_CAR','FLAG_OWN_CAR_CODE']].head(20))
3 print(normalised_home_loan['FLAG_OWN_CAR_CODE'].value_counts())
```

	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 [112]: 1 normalised_home_loan['CNT_CHILDREN_CODE']=ordenc.fit_transform(normalised_home_loan[['CNT_CHILDREN_CODE','CNT_CHILDREN']].head(20))
2 print(normalised_home_loan[['CNT_CHILDREN_CODE','CNT_CHILDREN']].head(20))
3 print(normalised_home_loan['CNT_CHILDREN_CODE'].value_counts())
```

	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	
7.0	1	
9.0	1	
10.0	1	

Name: CNT\_CHILDREN\_CODE, dtype: int64

```
In [115]: 1 normalised_home_loan=normalised_home_loan.sample(frac=1,random_state=45)
```

```
In [114]: 1 normalised_home_loan['TARGET'].value_counts()
```

```
Out[114]: 0    24825
1    24825
Name: TARGET, dtype: int64
```

```
In [116]: 1 y=normalised_home_loan.TARGET
```

```
In [117]: 1 normalised_home_loan_features=['SK_ID_CURR','NAME_CONTRACT_TYPE_CODE','CNT_C
```

```
In [118]: 1 from sklearn.model_selection import train_test_split
```

```
In [119]: 1 X=normalised_home_loan[normalised_home_loan_features]
```

```
In [ ]: 1 blobs_random_seed = 42
2 centers = [(0,0), (5,5)]
3 cluster_std = 1
4 frac_test_split = 0.33
5 num_features_for_samples = 2
6 num_samples_total = 49650
7
8 # Generate data
9 inputs, targets = make_blobs(n_samples = num_samples_total, centers = center
10
11 X_train,X_test,y_train,y_test=train_test_split(inputs,targets,test_size=0.33
```

```
In [120]: 1 print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

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

```
In [121]: 1 plt.pyplot.scatter(X_train[:,0], X_train[:,1])
2 plt.pyplot.title('Linearly separable data')
3 plt.pyplot.xlabel('X1')
4 plt.pyplot.ylabel('X2')
5 plt.pyplot.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-121-b6594a5f9cf2> in <module>
----> 1 plt.pyplot.scatter(X_train[:,0], X_train[:,1])
      2 plt.pyplot.title('Linearly separable data')
      3 plt.pyplot.xlabel('X1')
      4 plt.pyplot.ylabel('X2')
      5 plt.pyplot.show()
```

```
AttributeError: module 'matplotlib.pyplot' has no attribute 'pyplot'
```

```
In [122]: 1 from sklearn import svm
2 from sklearn.metrics import plot_confusion_matrix
```

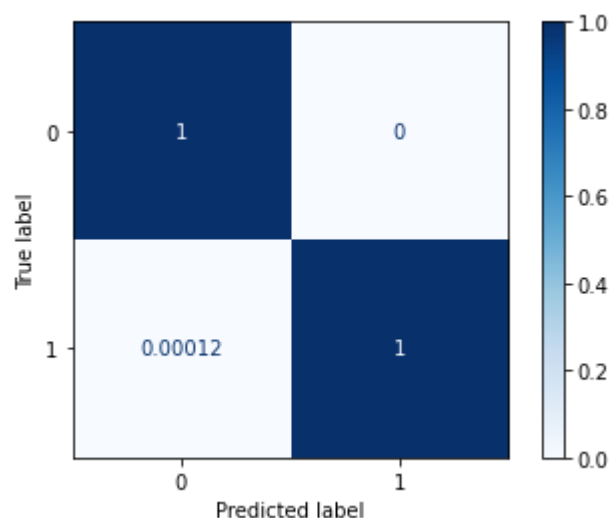
```
In [123]: 1 clf=svm.SVC(kernel='linear')
```

```
In [125]: 1 clf=clf.fit(X_train,y_train)
```

```
In [126]: 1 predictions = clf.predict(X_test)
2
3 # Generate confusion matrix
4 matrix = plot_confusion_matrix(clf, X_test, y_test,
5                               cmap=plt.cm.Blues,
6                               normalize='true')
7 plt.pyplot.title('Confusion matrix for our classifier')
8 plt.pyplot.show(matrix)
9 plt.pyplot.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-126-6eebb7ca2459> in <module>
5                                cmap=plt.cm.Blues,
6                                normalize='true')
----> 7 plt.pyplot.title('Confusion matrix for our classifier')
8     plt.pyplot.show(matrix)
9     plt.pyplot.show()
```

**AttributeError:** module 'matplotlib.pyplot' has no attribute 'pyplot'



```
In [127]: 1 from sklearn.metrics import precision_score, recall_score, f1_score
```

```
In [128]: 1 print(precision_score(y_test, predictions))
          2 print(recall_score(y_test, predictions))
          3 print(f1_score(y_test, predictions, average=None))

1.0
0.9998781676413255
[0.99993886 0.99993908]
```

```
In [129]: 1 support_vectors = clf.support_vectors_
          2
          3 # Visualize support vectors
          4 plt.pyplot.scatter(X_train[:,0], X_train[:,1])
          5 plt.pyplot.scatter(support_vectors[:,0], support_vectors[:,1], color='red')
          6 plt.pyplot.title('Linearly separable data with support vectors')
          7 plt.pyplot.xlabel('X1')
          8 plt.pyplot.ylabel('X2')
          9 plt.pyplot.show()
         10
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-129-f3a5707ad216> in <module>
      2
      3 # Visualize support vectors
----> 4 plt.pyplot.scatter(X_train[:,0], X_train[:,1])
      5 plt.pyplot.scatter(support_vectors[:,0], support_vectors[:,1], color='red')
      6 plt.pyplot.title('Linearly separable data with support vectors')

AttributeError: module 'matplotlib.pyplot' has no attribute 'pyplot'
```

```
In [132]: 1 from mlxtend.plotting import plot_decision_regions
```

```
-----
ModuleNotFoundError                            Traceback (most recent call last)
<ipython-input-132-4e4b181d624c> in <module>
----> 1 from mlxtend.plotting import plot_decision_regions

ModuleNotFoundError: No module named 'mlxtend'
```

```
In [131]: 1 plot_decision_regions(X_test, y_test, clf=clf, legend=2)
          2 plt.pyplot.show()
```

```
-----
NameError                                    Traceback (most recent call last)
<ipython-input-131-ead453bdfdbf> in <module>
----> 1 plot_decision_regions(X_test, y_test, clf=clf, legend=2)
      2 plt.pyplot.show()

NameError: name 'plot_decision_regions' is not defined
```

In [ ]:

1