In [1]:
```
import pandas as pd
```

In [2]:
```
#from google.colab import files
#data_to_load = files.upload()
```

In [3]:
```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [4]:
```
train= pd.read_csv('Train.csv')
test= pd.read_csv('Test.csv')

pd.set_option('max_column', None)
pd.set_option('max_row', None)

train.head()
```

Out[4]:

| | Applicant_ID | form_field1 | form_field2 | form_field3 | form_field4 | form_field5 | form_field6 | form_f |
|---|---|---|---|---|---|---|---|---|
| **0** | Apcnt_1000000 | 3436.0 | 0.28505 | 1.6560 | 0.0 | 0.000 | 0.0 | 10689 |
| **1** | Apcnt_1000004 | 3456.0 | 0.67400 | 0.2342 | 0.0 | 0.000 | 0.0 | 898 |
| **2** | Apcnt_1000008 | 3276.0 | 0.53845 | 3.1510 | 0.0 | 6.282 | NaN | 956 |
| **3** | Apcnt_1000012 | 3372.0 | 0.17005 | 0.5050 | 0.0 | 0.000 | 192166.0 | 3044 |
| **4** | Apcnt_1000016 | 3370.0 | 0.77270 | 1.1010 | 0.0 | 0.000 | 1556.0 | 214 |

In [5]:
```
print('train data shape :', train.shape)
print('test data shape :', test.shape)
```

```
train data shape : (56000, 52)
test data shape : (24000, 51)
```

In [6]:
```
#pip install CatBoost
```

In [7]:
```
train['default_status'] = train['default_status'].replace({'yes':1, 'no':0})
train['form_field47'] = train['form_field47'].replace({'charge':0, 'lending':1})

test['form_field47'] = test['form_field47'].replace({'charge':0, 'lending':1})
```

dummy=pd.get_dummies(train['form_field47']) dummy=pd.get_dummies(test['form_field47'])

train=pd.concat([train, dummy], axis=1) test=pd.concat([test, dummy], axis=1)

train=train.merge(dummy,left_index=True, right_index=True)
test=test.merge(dummy,left_index=True, right_index=True)

train=train.drop(['form_field47'], axis=1) test=test.drop(['form_field47'], axis=1)

train['new_df']=(train.form_field3+train.form_field4+train.form_field5)/3 test['new_df']= (test.form_field3+test.form_field4+test.form_field5)/3

train.head()

In [8]:
```
test_df = pd.DataFrame({ 'Applicant_ID':test['Applicant_ID']})
```

In [9]:
```
data= train.append(test)
```

In [10]:
```
corr_col = list(data.corr()[abs(data.corr()) >= .8].dropna(thresh = 2).index)
data_cor = data[corr_col].corr().loc[corr_col,corr_col]
```

In [11]:
```
data= data.drop(['Applicant_ID'], axis =1)
default = data['default_status'].copy()
```

In [12]:
```
same =[]
for col in corr_col:
    cor = (list(data_cor[col][np.abs(data_cor[col]) >=.8].keys()))
    same.append((cor))
```

In [13]:
```
col_ = 51
for col in same:
    data['form_field' + str(col_)] = data.loc[:, col].std(axis = 1)
    col_ += 1
```

In [14]:
```
nan_col = []
for i in data.drop(columns =['default_status']).columns:
    if (data[i].isnull().sum()) / len(data) * 100 > 70:
        nan_col.append(i)
```

In [15]:
```
#data[nan_col].fillna(data[nan_col].mean(), inplace=True)
```

In [16]:
```
from sklearn.preprocessing import scale
data_df = data.drop(columns = ['default_status']).fillna(-3333)
data_df = pd.DataFrame(data= scale(data_df),columns=data_df.columns)
data_df['default_status']= default.values
```

In [17]:
```
train = data_df[data_df['default_status'].isna()== False]
test = data_df[data_df['default_status'].isna()== True]

test.drop('default_status', axis=1, inplace=True)
test.reset_index(inplace=True, drop=True)
print(test.shape, train.shape)
```

```
(24000, 65) (56000, 66)
```

In [18]:
```
X = train.drop('default_status', axis=1)
y = train.default_status
```

In [19]:
```
from catboost import CatBoostClassifier
cat = CatBoostClassifier(iterations=10000,early_stopping_rounds=300,random_state=
                         learning_rate= 0.01, verbose= 1000)
```

In [20]:
```
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score
kfold, scores = KFold(n_splits=5, shuffle=True, random_state=221), list()
pred_tot_cb = []
for train_, test_ in kfold.split(X):
    x_train, x_test = X.iloc[train_], X.iloc[test_]
    y_train, y_test = y.iloc[train_], y.iloc[test_]

    model = cat
    model.fit(x_train, y_train, eval_set = (x_test, y_test))
    preds = model.predict_proba(x_test)
    score = roc_auc_score(y_test, preds[:, 1])
    scores.append(score)
    test_pred = model.predict_proba(test)[:,1]
    pred_tot_cb.append(test_pred)
    print(score, '\n\n')
```

```
0:      test: 0.7917810 best: 0.7917810 (0)     total: 95.6ms   remaining: 15
m 55s
1000:   test: 0.8367630 best: 0.8367630 (1000)  total: 40.2s    remaining: 6m
1s
2000:   test: 0.8384450 best: 0.8384544 (1972)  total: 1m 19s   remaining: 5m
17s
Stopped by overfitting detector  (300 iterations wait)

bestTest = 0.8385872288
bestIteration = 2459

Shrink model to first 2460 iterations.
0.8385872287750675


0:      test: 0.7774174 best: 0.7774174 (0)     total: 43.8ms   remaining: 7m
17s
1000:   test: 0.8383794 best: 0.8383898 (995)   total: 40.2s    remaining: 6m
1s
2000:   test: 0.8410303 best: 0.8410325 (1999)  total: 1m 19s   remaining: 5m
18s
3000:   test: 0.8413911 best: 0.8414178 (2965)  total: 1m 58s   remaining: 4m
37s
Stopped by overfitting detector  (300 iterations wait)

bestTest = 0.8414177859
bestIteration = 2965

Shrink model to first 2966 iterations.
0.8414177858688616


0:      test: 0.7859607 best: 0.7859607 (0)     total: 43.8ms   remaining: 7m
17s
1000:   test: 0.8406249 best: 0.8406249 (1000)  total: 40.1s    remaining: 6m
Stopped by overfitting detector  (300 iterations wait)

bestTest = 0.8416988728
bestIteration = 1689
```

```
Shrink model to first 1690 iterations.
0.8416988727801932


0:      test: 0.7785323 best: 0.7785323 (0)     total: 43ms     remaining: 7m
10s
1000:   test: 0.8393501 best: 0.8393501 (1000)  total: 40s      remaining: 5m
59s
2000:   test: 0.8414283 best: 0.8414627 (1985)  total: 1m 19s   remaining: 5m
17s
3000:   test: 0.8420493 best: 0.8420916 (2854)  total: 1m 58s   remaining: 4m
36s
Stopped by overfitting detector  (300 iterations wait)

bestTest = 0.842152964
bestIteration = 3169

Shrink model to first 3170 iterations.
0.8421529639591178


0:      test: 0.7906311 best: 0.7906311 (0)     total: 42.6ms   remaining: 7m
5s
1000:   test: 0.8370343 best: 0.8370343 (1000)  total: 40.5s    remaining: 6m
4s
2000:   test: 0.8388719 best: 0.8388719 (2000)  total: 1m 24s   remaining: 5m
37s
Stopped by overfitting detector  (300 iterations wait)

bestTest = 0.8390966625
bestIteration = 2328

Shrink model to first 2329 iterations.
0.839096662466081
```

In [21]:
```
sum(scores) / len(scores)
```

Out[21]:
```
0.8405907027698643
```

In [22]:
```
prediction = sum(pred_tot_cb)/len(pred_tot_cb)
```

In [23]:
```
submit = pd.read_csv('SampleSubmission.csv')
submit.head()
```

Out[23]:

| | Applicant_ID | default_status |
|---|---|---|
| **0** | Apcnt_1000032 | 1 |
| **1** | Apcnt_1000048 | 1 |
| **2** | Apcnt_1000052 | 1 |
| **3** | Apcnt_1000076 | 1 |
| **4** | Apcnt_1000080 | 1 |

In [24]:
```
pd.DataFrame({'Applicant_ID':submit.Applicant_ID,'default_status':prediction}).to
print('Donee')
```

```
Donee
```

In [1]:
```
pip install -U notebook-as-pdf
```

```
Note: you may need to restart the kernel to use updated packages.

'C:\Users\Seun' is not recognized as an internal or external command,
operable program or batch file.
```

In [ ]:
```

```