Importing the necessary libraries

In [103]:

```
import pandas as pd
import numpy as np
import math
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import VotingClassifier
from sklearn.pipeline import Pipeline
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import style
style.use('ggplot')
```

In [104]:

```
!pip install catboost
```

```
Requirement already satisfied: catboost in /usr/local/lib/python3.6/dist-packag
es (0.24.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-pack
ages (from catboost) (3.2.2)
Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages
(from catboost) (4.4.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages
(from catboost) (1.4.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.6/dist-p
ackages (from catboost) (1.18.5)
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.6/dist-
packages (from catboost) (1.1.2)
Requirement already satisfied: graphviz in /usr/local/lib/python3.6/dist-packag
es (from catboost) (0.10.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (f
rom catboost) (1.15.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /us
r/local/lib/python3.6/dist-packages (from matplotlib->catboost) (2.4.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-pa
ckages (from matplotlib->catboost) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.
6/dist-packages (from matplotlib->catboost) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/di
st-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist
-packages (from plotly->catboost) (1.3.3)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-pa
ckages (from pandas>=0.24.0->catboost) (2018.9)
```

In [105]:

```
import xgboost as xgb
from xgboost import plot_tree
from catboost import CatBoostClassifier
```

Reading in the train, test and samplesubmission files

In [106]:

```
train = pd.read_csv('/Train(1).csv')
test = pd.read_csv('/Test(1).csv')
sub = pd.read_csv('/SampleSubmission(1).csv')
```

In [107]:

```
Train = train.copy()
Train = Train.set_index(['Applicant_ID'])
Train.head()
```

Out[107]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000000	3436.0	0.28505	1.6560	0.0	0.000	0.0	10689720
Apcnt_1000004	3456.0	0.67400	0.2342	0.0	0.000	0.0	898979
Apcnt_1000008	3276.0	0.53845	3.1510	0.0	6.282	NaN	956940
Apcnt_1000012	3372.0	0.17005	0.5050	0.0	0.000	192166.0	3044703
Apcnt_1000016	3370.0	0.77270	1.1010	0.0	0.000	1556.0	214728

Understanding the data set(train and test)

In [108]:

```
Train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 56000 entries, Apcnt_1000000 to Apcnt_999996
Data columns (total 51 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   form_field1  53471 non-null  float64
 1   form_field2  52156 non-null  float64
 2   form_field3  55645 non-null  float64
 3   form_field4  55645 non-null  float64
 4   form_field5  55645 non-null  float64
 5   form_field6  42640 non-null  float64
 6   form_field7  50837 non-null  float64
 7   form_field8  42640 non-null  float64
 8   form_field9  47992 non-null  float64
 9   form_field10  55645 non-null  float64
 10  form_field11  24579 non-null  float64
 11  form_field12  46105 non-null  float64
 12  form_field13  50111 non-null  float64
 13  form_field14  56000 non-null  int64  
 14  form_field15  33525 non-null  float64
 15  form_field16  42964 non-null  float64
 16  form_field17  44849 non-null  float64
 17  form_field18  45598 non-null  float64
 18  form_field19  55996 non-null  float64
 19  form_field20  55645 non-null  float64
 20  form_field21  40146 non-null  float64
 21  form_field22  35600 non-null  float64
 22  form_field23  27877 non-null  float64
 23  form_field24  42703 non-null  float64
 24  form_field25  50550 non-null  float64
 25  form_field26  48562 non-null  float64
 26  form_field27  46701 non-null  float64
 27  form_field28  55645 non-null  float64
 28  form_field29  55645 non-null  float64
 29  form_field30  30491 non-null  float64
 30  form_field31  16592 non-null  float64
 31  form_field32  50550 non-null  float64
 32  form_field33  54744 non-null  float64
 33  form_field34  55645 non-null  float64
 34  form_field35  32852 non-null  float64
 35  form_field36  54005 non-null  float64
 36  form_field37  50550 non-null  float64
 37  form_field38  55645 non-null  float64
 38  form_field39  51789 non-null  float64
 39  form_field40  12271 non-null  float64
 40  form_field41  17771 non-null  float64
 41  form_field42  54677 non-null  float64
 42  form_field43  55432 non-null  float64
 43  form_field44  50617 non-null  float64
 44  form_field45  24683 non-null  float64
 45  form_field46  40096 non-null  float64
 46  form_field47  56000 non-null  object 
 47  form_field48  35111 non-null  float64
 48  form_field49  55645 non-null  float64
 49  form_field50  44944 non-null  float64
 50  default_status  56000 non-null  object
dtypes: float64(48), int64(1), object(2)
memory usage: 22.2+ MB
```

In [109]:
```
Train.describe()
```

Out[109]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	forr
count	53471.000000	52156.000000	55645.000000	55645.000000	55645.000000	4.264000e+04	5.0837
mean	3491.795665	0.550737	1.052225	0.851979	1.956317	6.244479e+05	6.8652
std	188.462426	0.820979	2.147768	3.157692	10.512396	1.433422e+06	1.9127
min	2990.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.0000
25%	3358.000000	0.070788	0.000000	0.000000	0.000000	1.400400e+04	6.8697
50%	3484.000000	0.267575	0.062000	0.000000	0.000000	1.155330e+05	2.7043
75%	3620.000000	0.719512	1.282000	0.000000	0.000000	5.259280e+05	6.9938
max	3900.000000	18.015050	57.371600	91.672200	407.748600	5.313546e+07	2.1587

In [110]:
```
Train.describe(include = 'all')
```

Out[110]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	fo
count	53471.000000	52156.000000	55645.000000	55645.000000	55645.000000	4.264000e+04	5.083
unique	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	
mean	3491.795665	0.550737	1.052225	0.851979	1.956317	6.244479e+05	6.865
std	188.462426	0.820979	2.147768	3.157692	10.512396	1.433422e+06	1.912
min	2990.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000
25%	3358.000000	0.070788	0.000000	0.000000	0.000000	1.400400e+04	6.869
50%	3484.000000	0.267575	0.062000	0.000000	0.000000	1.155330e+05	2.704
75%	3620.000000	0.719512	1.282000	0.000000	0.000000	5.259280e+05	6.993
max	3900.000000	18.015050	57.371600	91.672200	407.748600	5.313546e+07	2.158

Finding numeric and categorical columns

In [111]:
```
numeric = Train.select_dtypes(exclude = 'object')
numeric.columns
```

Out[111]:
```
Index(['form_field1', 'form_field2', 'form_field3', 'form_field4',
       'form_field5', 'form_field6', 'form_field7', 'form_field8',
       'form_field9', 'form_field10', 'form_field11', 'form_field12',
       'form_field13', 'form_field14', 'form_field15', 'form_field16',
       'form_field17', 'form_field18', 'form_field19', 'form_field20',
       'form_field21', 'form_field22', 'form_field23', 'form_field24',
       'form_field25', 'form_field26', 'form_field27', 'form_field28',
       'form_field29', 'form_field30', 'form_field31', 'form_field32',
       'form_field33', 'form_field34', 'form_field35', 'form_field36',
       'form_field37', 'form_field38', 'form_field39', 'form_field40',
       'form_field41', 'form_field42', 'form_field43', 'form_field44',
       'form_field45', 'form_field46', 'form_field48', 'form_field49',
       'form_field50'],
      dtype='object')
```

In [112]:
```
objects = Train.select_dtypes(include = 'object')
objects.columns
```

Out[112]:
```
Index(['form_field47', 'default_status'], dtype='object')
```

In [113]:
```
Train.isnull().sum()
```

Out[113]:
```
form_field1        2529
form_field2        3844
form_field3         355
form_field4         355
form_field5         355
form_field6       13360
form_field7        5163
form_field8       13360
form_field9        8008
form_field10        355
form_field11      31421
form_field12       9895
form_field13       5889
form_field14          0
form_field15      22475
form_field16      13036
form_field17      11151
form_field18      10402
form_field19          4
form_field20        355
```

In [114]:
```
Train.shape
```

Out[114]:
```
(56000, 51)
```

In [115]:
```
Train['form_field47'].value_counts()
```

Out[115]:
```
charge     36373
lending    19627
Name: form_field47, dtype: int64
```

In [116]:
```
Test = test.set_index(['Applicant_ID'])
Test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 24000 entries, Apcnt_1000032 to Apcnt_999992
Data columns (total 50 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   form_field1   22890 non-null  float64
 1   form_field2   22291 non-null  float64
 2   form_field3   23854 non-null  float64
 3   form_field4   23854 non-null  float64
 4   form_field5   23854 non-null  float64
 5   form_field6   18396 non-null  float64
 6   form_field7   21769 non-null  float64
 7   form_field8   18396 non-null  float64
 8   form_field9   20600 non-null  float64
 9   form_field10  23853 non-null  float64
 10  form_field11  10602 non-null  float64
 11  form_field12  19817 non-null  float64
 12  form_field13  21537 non-null  float64
 13  form_field14  24000 non-null  int64  
 14  form_field15  14408 non-null  float64
 15  form_field16  18526 non-null  float64
 16  form_field17  19305 non-null  float64
 17  form_field18  19631 non-null  float64
 18  form_field19  24000 non-null  float64
 19  form_field20  23853 non-null  float64
 20  form_field21  17293 non-null  float64
 21  form_field22  15276 non-null  float64
 22  form_field23  11875 non-null  float64
 23  form_field24  18395 non-null  float64
 24  form_field25  21744 non-null  float64
 25  form_field26  20828 non-null  float64
 26  form_field27  20090 non-null  float64
 27  form_field28  23853 non-null  float64
 28  form_field29  23853 non-null  float64
 29  form_field30  13092 non-null  float64
 30  form_field31  7190 non-null   float64
 31  form_field32  21744 non-null  float64
 32  form_field33  23505 non-null  float64
 33  form_field34  23853 non-null  float64
 34  form_field35  14134 non-null  float64
 35  form_field36  23097 non-null  float64
 36  form_field37  21744 non-null  float64
 37  form_field38  23853 non-null  float64
 38  form_field39  22171 non-null  float64
 39  form_field40  5172 non-null   float64
 40  form_field41  7651 non-null   float64
 41  form_field42  23422 non-null  float64
 42  form_field43  23750 non-null  float64
 43  form_field44  21638 non-null  float64
 44  form_field45  10462 non-null  float64
 45  form_field46  17115 non-null  float64
 46  form_field47  24000 non-null  object 
 47  form_field48  15078 non-null  float64
 48  form_field49  23854 non-null  float64
 49  form_field50  19203 non-null  float64
dtypes: float64(48), int64(1), object(1)
memory usage: 9.3+ MB
```

In [117]:
```
Test.isnull().sum()
```

Out[117]:
```
form_field1      1110
form_field2      1709
form_field3       146
form_field4       146
form_field5       146
form_field6      5604
form_field7      2231
form_field8      5604
form_field9      3400
form_field10      147
form_field11    13398
form_field12     4183
form_field13     2463
form_field14        0
form_field15     9592
form_field16     5474
form_field17     4695
form_field18     4369
form_field19        0
form_field20      147
```

In [118]:
```
Test['form_field47'].value_counts()
```

Out[118]:
```
charge     15467
lending     8533
Name: form_field47, dtype: int64
```

Preprocessing and cleaning of the data set(train and test)

In [119]:
```
Train.replace('no', 0, inplace = True)
Train.replace('yes', 1, inplace = True)
Train.head(3)
```

Out[119]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000000	3436.0	0.28505	1.6560	0.0	0.000	0.0	10689720
Apcnt_1000004	3456.0	0.67400	0.2342	0.0	0.000	0.0	898979
Apcnt_1000008	3276.0	0.53845	3.1510	0.0	6.282	NaN	956940

In [120]:
```
Train = pd.get_dummies(Train)
Train.head()
```

Out[120]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000000	3436.0	0.28505	1.6560	0.0	0.000	0.0	1068972
Apcnt_1000004	3456.0	0.67400	0.2342	0.0	0.000	0.0	89897
Apcnt_1000008	3276.0	0.53845	3.1510	0.0	6.282	NaN	95694
Apcnt_1000012	3372.0	0.17005	0.5050	0.0	0.000	192166.0	304470
Apcnt_1000016	3370.0	0.77270	1.1010	0.0	0.000	1556.0	21472

In [121]:
```
Test = pd.get_dummies(Test)
Test.head(3)
```

Out[121]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000032	3236.0	0.34875	10.2006	0.0000	0.0	418564.0	41856
Apcnt_1000048	3284.0	1.27360	2.9606	9.0198	0.0	0.0	985881
Apcnt_1000052	NaN	0.27505	0.0600	0.0000	0.0	NaN	N

In [122]:
```
Train.fillna(-999, inplace = True)
Train.head()
```

Out[122]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000000	3436.0	0.28505	1.6560	0.0	0.000	0.0	1068972
Apcnt_1000004	3456.0	0.67400	0.2342	0.0	0.000	0.0	89897
Apcnt_1000008	3276.0	0.53845	3.1510	0.0	6.282	-999.0	95694
Apcnt_1000012	3372.0	0.17005	0.5050	0.0	0.000	192166.0	304470
Apcnt_1000016	3370.0	0.77270	1.1010	0.0	0.000	1556.0	21472

In [123]:
```
Test.fillna(-999, inplace = True)
Test.head()
```

Out[123]:

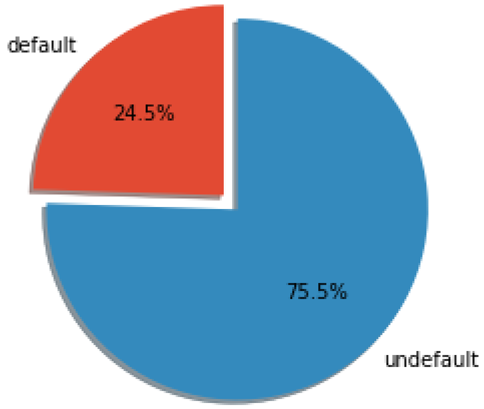
	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_fiel
Applicant_ID							
Apcnt_1000032	3236.0	0.34875	10.2006	0.0000	0.0	418564.0	418564
Apcnt_1000048	3284.0	1.27360	2.9606	9.0198	0.0	0.0	9858816
Apcnt_1000052	-999.0	0.27505	0.0600	0.0000	0.0	-999.0	-999
Apcnt_1000076	3232.0	0.28505	2.8032	0.0000	0.0	0.0	473802
Apcnt_1000080	3466.0	2.09545	0.8318	2.5182	0.0	19839.0	1150662

Visualizing and Exploring the train set

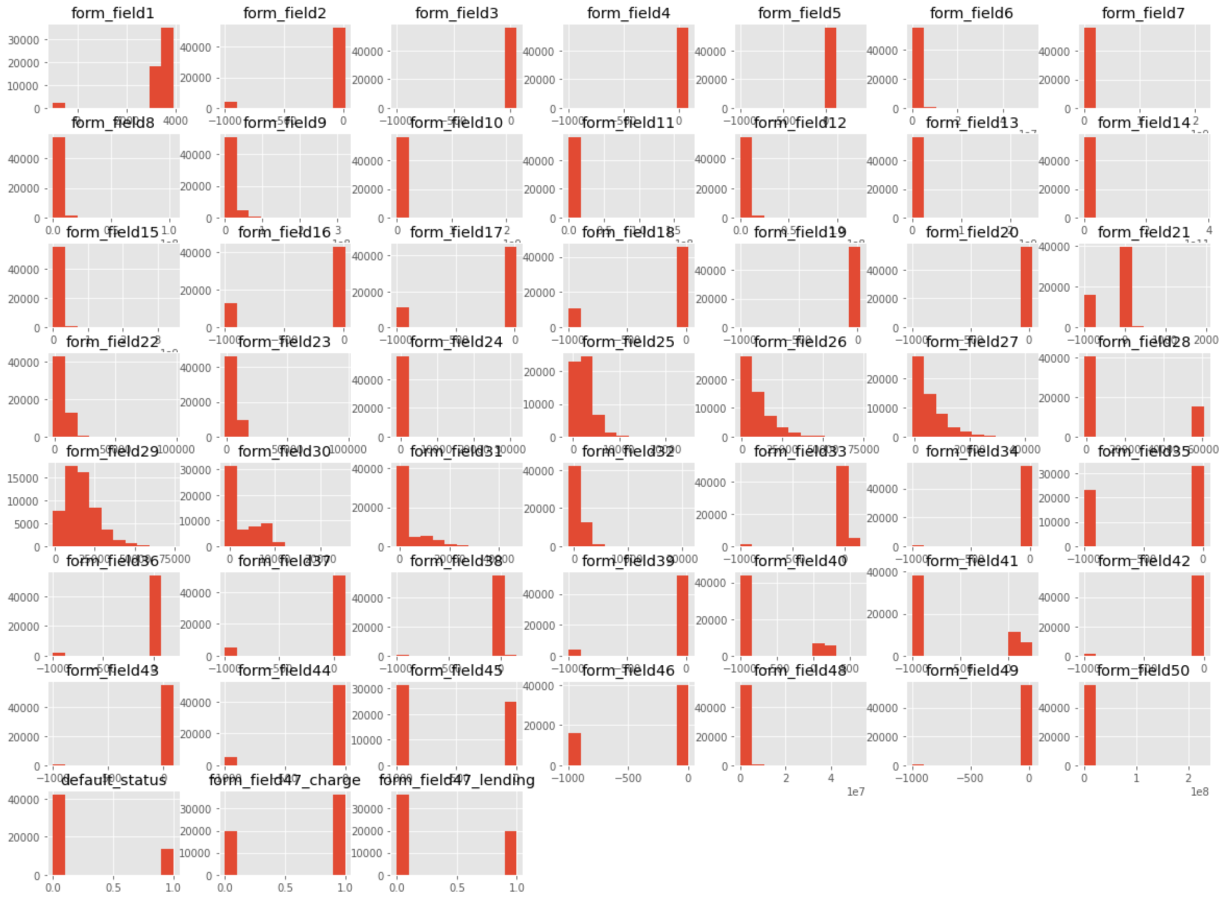
In [124]:

```python
labels = 'default', 'undefault'
default_size = 0
undefault_size = 0
for i in Train['default_status']:
    if i == 1:
        default_size += 1
    else:
        undefault_size +=1
print(default_size)
print(undefault_size)
sizes = [(default_size / len(Train)) * 100, (undefault_size / len(Train)) * 100]
explode = (0.1, 0)  # only "explode" the 1st slice (i.e. 'default')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```

```
13715
42285
```



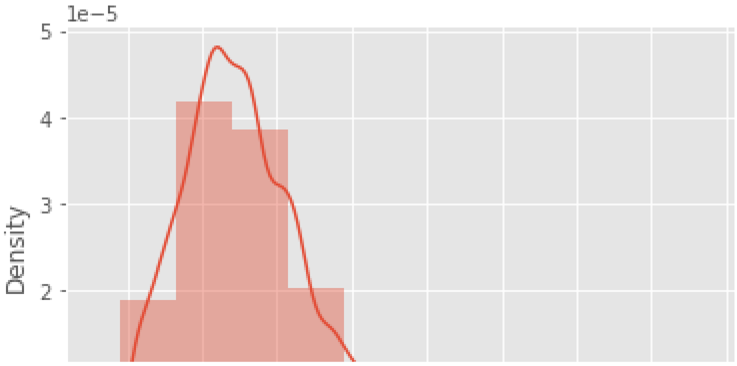
In [125]:
```
Train.hist(bins=10, figsize=(20, 15))
plt.show()
```



```
In [126]: sns.distplot(Train['form_field29'], bins=10)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureW
arning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level functi
on with similar flexibility) or `histplot` (an axes-level function for histog
rams).
  warnings.warn(msg, FutureWarning)
```

```
Out[126]: <matplotlib.axes._subplots.AxesSubplot at 0x7f496c608240>
```



Determining correlation between target and features

In [127]:

```
corrMatrix = Train.corr()
corrMatrix
```

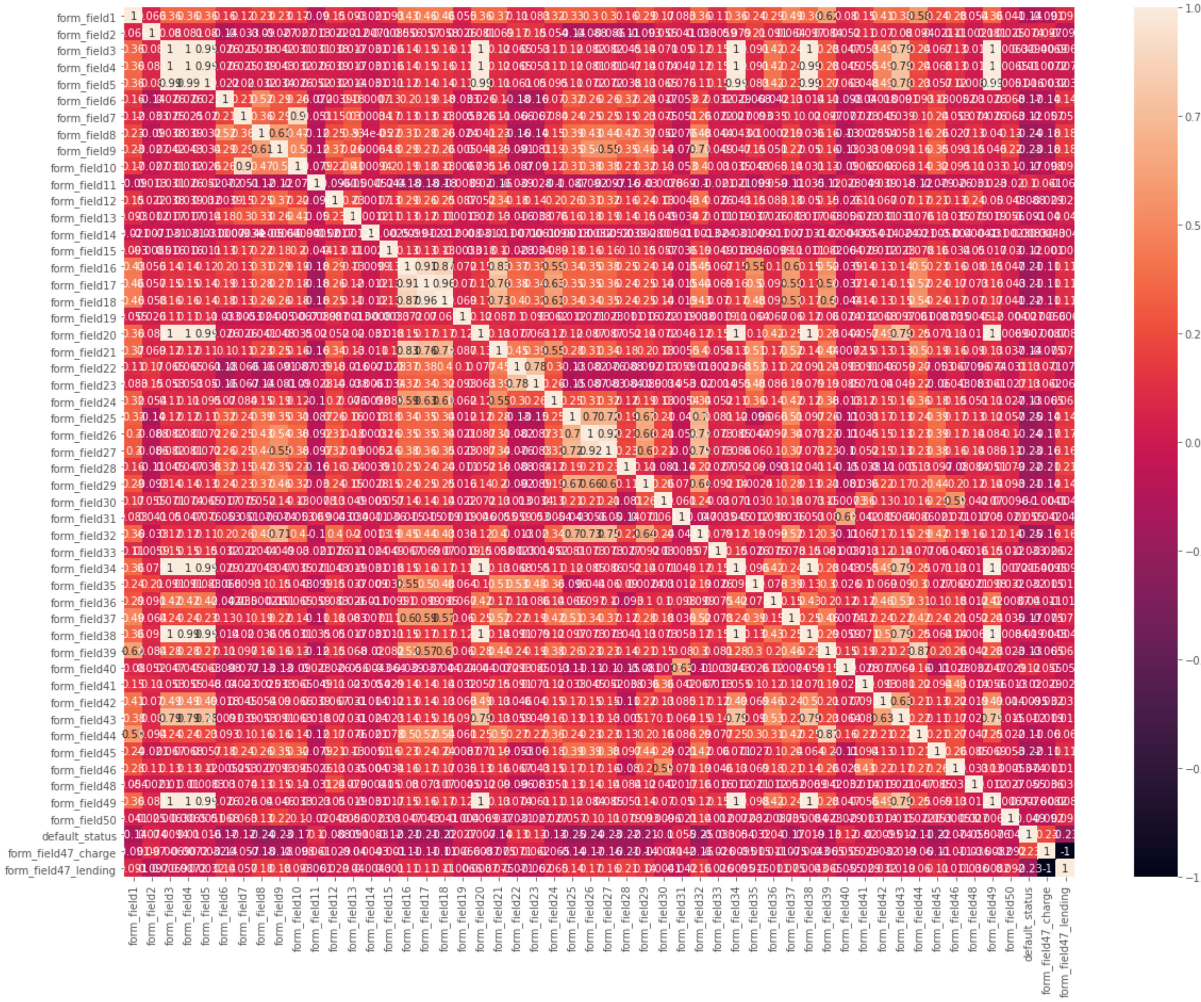
Out[127]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	for
form_field1	1.000000	0.066395	0.358717	0.359764	0.355599	0.155828	0
form_field2	0.066395	1.000000	0.080174	0.080579	0.080074	-0.138547	-0
form_field3	0.358717	0.080174	1.000000	0.999254	0.993343	0.025861	0
form_field4	0.359764	0.080579	0.999254	1.000000	0.991149	0.026420	0
form_field5	0.355599	0.080074	0.993343	0.991149	1.000000	0.021725	0
form_field6	0.155828	-0.138547	0.025861	0.026420	0.021725	1.000000	0
form_field7	0.123642	-0.032814	0.024715	0.025235	0.020352	0.207660	1
form_field8	0.230769	-0.090430	0.038259	0.039185	0.032050	0.522087	0
form_field9	0.229502	-0.026720	0.041953	0.042613	0.034483	0.289385	0
form_field10	0.172198	-0.027382	0.031477	0.032431	0.025820	0.258334	0
form_field11	-0.089929	0.012688	0.030722	0.025518	0.052391	-0.071671	-0
form_field12	0.150727	0.022158	0.038149	0.038887	0.031910	0.003930	0
form_field13	0.093389	-0.012013	0.016790	0.017195	0.013911	0.183866	0
form_field14	-0.021031	-0.007145	-0.030710	-0.030709	-0.030823	-0.000700	0
form_field15	0.092517	-0.008544	0.015772	0.016036	0.011432	0.132512	0
form_field16	0.431777	0.055683	0.136770	0.137795	0.121704	0.204917	0
form_field17	0.458290	0.057080	0.152245	0.152967	0.135747	0.185440	0
form_field18	0.460704	0.057772	0.159336	0.159954	0.142585	0.177611	0
form_field19	0.054589	0.026108	0.111975	0.111989	0.110062	-0.033103	-0
form_field20	0.361945	0.081049	0.999346	0.998931	0.991007	0.025788	0
form_field21	0.371567	0.069337	0.119697	0.120604	0.106645	0.104013	0
form_field22	0.108731	0.166758	0.065471	0.065385	0.061349	-0.178472	-0
form_field23	0.083393	0.145292	0.053084	0.052822	0.049722	-0.162650	-0
form_field24	0.322807	0.054365	0.107872	0.106595	0.094787	0.069625	0
form_field25	0.332689	-0.140681	0.118576	0.117680	0.108797	0.318613	0
form_field26	0.302250	-0.088207	0.082369	0.081276	0.072006	0.255407	0
form_field27	0.303718	-0.086130	0.082392	0.081334	0.071907	0.255463	0
form_field28	0.155923	-0.113445	0.045485	0.047075	0.037931	0.320936	0
form_field29	0.291305	-0.092637	0.140925	0.140412	0.134865	0.237581	0
form_field30	0.171175	0.055030	0.070512	0.073910	0.064811	0.017363	0
form_field31	0.082676	0.041341	0.049570	0.046715	0.075906	-0.052720	-0
form_field32	0.355534	-0.032589	0.118484	0.117139	0.106186	0.199499	0
form_field33	0.106149	0.005889	0.150281	0.150386	0.147800	0.031733	0
form_field34	0.360407	0.079351	0.999621	0.999212	0.991395	0.028987	0

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	for
form_field35	0.241815	0.206719	0.091056	0.091366	0.082693	-0.067531	(
form_field36	0.285131	0.091039	0.417024	0.416246	0.416548	-0.042113	(
form_field37	0.491966	0.063798	0.241281	0.240333	0.229409	0.125746	(
form_field38	0.364180	0.097107	0.995056	0.994600	0.986949	0.013946	(
form_field39	0.622855	0.083604	0.276831	0.277598	0.268839	0.106078	(
form_field40	0.079766	0.052440	0.046772	0.045198	0.062885	-0.098012	-(
form_field41	0.150026	0.113881	0.052861	0.054992	0.048353	-0.039541	(
form_field42	0.413058	0.070366	0.491087	0.490614	0.488447	0.018392	(
form_field43	0.381166	0.080117	0.787299	0.786950	0.780580	0.009114	(
form_field44	0.579925	0.094005	0.241327	0.242454	0.232281	0.092688	(
form_field45	0.240394	-0.021014	0.067352	0.067842	0.056543	0.179975	(
form_field46	0.281385	0.105363	0.127518	0.133351	0.121485	-0.005218	(
form_field48	0.054497	-0.002065	0.010087	0.010339	0.008272	0.030454	(
form_field49	0.361472	0.080659	0.999499	0.998938	0.992554	0.025554	(
form_field50	0.040527	0.024883	0.006288	0.006456	0.005129	0.067747	(
default_status	-0.140965	0.073846	0.009359	0.009983	0.015713	-0.173197	-(
form_field47_charge	-0.090533	0.097002	-0.006916	-0.007203	-0.003183	-0.139724	-(
form_field47_lending	0.090533	-0.097002	0.006916	0.007203	0.003183	0.139724	(

In [128]:

```
plt.subplots(figsize = (20, 15))
sns.heatmap(corrMatrix, annot = True)
plt.show()
```



In [129]:
```
Train.columns
```

Out[129]:
```
Index(['form_field1', 'form_field2', 'form_field3', 'form_field4',
       'form_field5', 'form_field6', 'form_field7', 'form_field8',
       'form_field9', 'form_field10', 'form_field11', 'form_field12',
       'form_field13', 'form_field14', 'form_field15', 'form_field16',
       'form_field17', 'form_field18', 'form_field19', 'form_field20',
       'form_field21', 'form_field22', 'form_field23', 'form_field24',
       'form_field25', 'form_field26', 'form_field27', 'form_field28',
       'form_field29', 'form_field30', 'form_field31', 'form_field32',
       'form_field33', 'form_field34', 'form_field35', 'form_field36',
       'form_field37', 'form_field38', 'form_field39', 'form_field40',
       'form_field41', 'form_field42', 'form_field43', 'form_field44',
       'form_field45', 'form_field46', 'form_field48', 'form_field49',
       'form_field50', 'default_status', 'form_field47_charge',
       'form_field47_lending'],
      dtype='object')
```

Determining and Building the best model

In [130]:
```
X = np.array(Train.drop(['default_status'], 1))
y = np.array(Train['default_status'])
```

In [131]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, randon
```

In [132]:
```
print(Train.shape)
print(Test.shape)
```

```
(56000, 52)
(24000, 51)
```

xgboost classifier

In [133]:
```
xgb_clf = xgb.XGBClassifier(n_jobs = -1)
xgb_clf.fit(X_train, y_train)
preds1 = xgb_clf.predict_proba(X_test)[:,1]
accuracy1 = xgb_clf.score(X_test, y_test)
print(accuracy1)
```

```
0.808125
```

In [134]:
```
auc_df = roc_auc_score(y_test, preds1)
auc_df
```

Out[134]:
```
0.8310124691046903
```

catboost classifier

In [135]:
```
cat_clf = CatBoostClassifier(verbose=False)
cat_clf.fit(X_train, y_train)
preds3 = cat_clf.predict_proba(X_test)
accuracy3 = cat_clf.score(X_test, y_test)
print(accuracy3)
```

```
0.8091071428571428
```

In [136]:
```
auc_cat = roc_auc_score(y_test, preds3[:,1])
auc_cat
```

Out[136]:
```
0.8370318115528094
```

lightgbm classifier with manual parameter tuning

In [137]:
```
import lightgbm as lgb
lg_clf = lgb.LGBMClassifier(learning_rate=0.01, num_iterations=2500, scale_pos_we
lg_clf.fit(X_train, y_train)
preds4 = lg_clf.predict_proba(X_test)
accuracy4 = lg_clf.score(X_test, y_test)
print(accuracy4)
```

```
/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

0.746875
```

In [138]:
```
lg_cat = roc_auc_score(y_test, preds4[:,1])
lg_cat
```

Out[138]:
```
0.8362065544341879
```

randomforest classifier

In [139]:
```
from sklearn.ensemble import RandomForestClassifier
rf_clf = RandomForestClassifier()
rf_clf.fit(X_train, y_train)
preds5 = rf_clf.predict_proba(X_test)
accuracy5 = rf_clf.score(X_test, y_test)
print(accuracy5)
```

```
0.8046428571428571
```

In [140]:
```
rf_auc = roc_auc_score(y_test, preds5[:,1])
rf_auc
```

Out[140]:
```
0.8304646206478884
```

xgbclassifier tuned manually

In [141]:
```
xgb_clf = xgb.XGBClassifier(n_jobs = -1, scale_pos_weight=3, eta=0.01, n_estimato
xgb_clf.fit(X_train, y_train)
predstun = xgb_clf.predict_proba(X_test)
accuracytun = xgb_clf.score(X_test, y_test)
print(accuracytun)
```

```
0.7473214285714286
```

In [142]:
```
auc_df = roc_auc_score(y_test, predstun[:,1])
auc_df
```

Out[142]:
```
0.8353853787702872
```

Looking for the best model by predicting on the test set using stratified kfold.

In [143]:
```
X = Train.drop(['default_status'], 1)
y = Train['default_status']
```

In [144]:
```
X.shape
```

Out[144]:
```
(56000, 51)
```

In [145]:
```
Test.shape
```

Out[145]:
```
(24000, 51)
```

stratified kfold to predict on test data with catboost

In [146]:
```
#Test = Test.set_index(['Applicant_ID'])
```

In [147]:

```
skfold = StratifiedKFold(n_splits=25, random_state=0, shuffle=True)

scores = []
preds = []
i = 1
cat = CatBoostClassifier(verbose=False)

for train_split, test_split in skfold.split(X, y):
    Xtrain, Xtest, ytrain, ytest = X.iloc[train_split], X.iloc[test_split], y[tra
    cat.fit(Xtrain, ytrain)
    score = roc_auc_score(ytest, cat.predict_proba(Xtest)[:,1])
    pred = cat.predict_proba(Test)[:,1]
    scores.append(score)
    preds.append(pred)
    print('AUC Score for {} split:'.format(i), score)
    i+=1
print('Final score:', np.mean(scores))
Final_pred1 = np.mean(preds, axis = 0)
```

```
AUC Score for 1 split: 0.8325093613570085
AUC Score for 2 split: 0.8460747010405343
AUC Score for 3 split: 0.8353037479939949
AUC Score for 4 split: 0.8322645424582836
AUC Score for 5 split: 0.838814256872185
AUC Score for 6 split: 0.854275594899139
AUC Score for 7 split: 0.845760858311332
AUC Score for 8 split: 0.8234780245379718
AUC Score for 9 split: 0.8328738934617177
AUC Score for 10 split: 0.8365418629531844
AUC Score for 11 split: 0.847058088519635
AUC Score for 12 split: 0.8408471291817066
AUC Score for 13 split: 0.8301152894516022
AUC Score for 14 split: 0.8536191279451161
AUC Score for 15 split: 0.8484519458528436
AUC Score for 16 split: 0.8450243925033312
AUC Score for 17 split: 0.8364641264855514
AUC Score for 18 split: 0.8277250503307448
AUC Score for 19 split: 0.8424251824994425
AUC Score for 20 split: 0.8522328107984087
AUC Score for 21 split: 0.8429325293340185
AUC Score for 22 split: 0.8224178362034514
AUC Score for 23 split: 0.8517405443368352
AUC Score for 24 split: 0.8389847031159282
AUC Score for 25 split: 0.8515822004203116
Final score: 0.840380712034571
```

stratified kfold to predict test data with voting classifier(xgboost and catboost)

In [148]:

```
skfold = StratifiedKFold(n_splits=25, random_state=0, shuffle=True)

scores = []
preds = []
i = 1
vc = VotingClassifier(estimators= [('xg', xgb.XGBClassifier(n_jobs = -1, scale_po

for train_split, test_split in skfold.split(X, y):
    Xtrain, Xtest, ytrain, ytest = X.iloc[train_split], X.iloc[test_split], y[tra
    vc.fit(Xtrain, ytrain)
    score = roc_auc_score(ytest, vc.predict_proba(Xtest)[:,1])
    pred = vc.predict_proba(Test)[:,1]
    scores.append(score)
    preds.append(pred)
    print('AUC Score for {} split:'.format(i), score)
    i+=1
print('Final score:', np.mean(scores))
Final_pred2 = np.mean(preds, axis = 0)
```

```
AUC Score for 1 split: 0.8305842435850977
AUC Score for 2 split: 0.8461189194319338
AUC Score for 3 split: 0.8376020258494244
AUC Score for 4 split: 0.831605580576694
AUC Score for 5 split: 0.8394074304153509
AUC Score for 6 split: 0.8538118410381184
AUC Score for 7 split: 0.846544925885662
AUC Score for 8 split: 0.821440743041535
AUC Score for 9 split: 0.831855791962175
AUC Score for 10 split: 0.8372288657659056
AUC Score for 11 split: 0.8459227518664654
AUC Score for 12 split: 0.8419016781223643
AUC Score for 13 split: 0.8306409481676809
AUC Score for 14 split: 0.8521994185439039
AUC Score for 15 split: 0.8476699207957268
AUC Score for 16 split: 0.8449317559263172
AUC Score for 17 split: 0.8358048987514529
AUC Score for 18 split: 0.832837296778509
AUC Score for 19 split: 0.8422054399214097
AUC Score for 20 split: 0.8538841116421557
AUC Score for 21 split: 0.843866435290658
AUC Score for 22 split: 0.8211435446847609
AUC Score for 23 split: 0.8526098201234653
AUC Score for 24 split: 0.8402762293466213
AUC Score for 25 split: 0.8509735996527206
Final score: 0.8405227286866445
```

Stratified kfold to predict test data with voting classifier(lightgbm, catboost and randomforest).

In [149]:
```
skfold = StratifiedKFold(n_splits=25, random_state=0, shuffle=True)

scores = []
preds = []
i = 1
vc = VotingClassifier(estimators= [('lg_clf', lgb.LGBMClassifier(learning_rate=0.

for train_split, test_split in skfold.split(X, y):
    Xtrain, Xtest, ytrain, ytest = X.iloc[train_split], X.iloc[test_split], y[tra
    vc.fit(Xtrain, ytrain)
    score = roc_auc_score(ytest, vc.predict_proba(Xtest)[:,1])
    pred = vc.predict_proba(Test)[:,1]
    scores.append(score)
    preds.append(pred)
    print('AUC Score for {} split:'.format(i), score)
    i+=1
print('Final score:', np.mean(scores))
Final_pred3 = np.mean(preds, axis = 0)
```

```
/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 1 split: 0.8340958309606391

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 2 split: 0.8476902900726476

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 3 split: 0.8379741074355922

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 4 split: 0.8334951079360148

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 5 split: 0.8415622681230694

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: F
ound `num_iterations` in params. Will use it instead of argument
```

```
  warnings.warn("Found `{}` in params. Will use it instead of argument".forma
t(alias))

AUC Score for 6 split: 0.8556722489689566

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 7 split: 0.8477603924004763

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 8 split: 0.8228751445186451

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 9 split: 0.8326851564252558

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 10 split: 0.8385672809787579

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 11 split: 0.848005997679777

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 12 split: 0.8428420470960049

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 13 split: 0.8314068156822954

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 14 split: 0.8563055886785178
```

```
/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: F
ound `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".forma
t(alias))

AUC Score for 15 split: 0.8466519956180745

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 16 split: 0.8441820459542052

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 17 split: 0.8371352030841517

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 18 split: 0.8350142563383346

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 19 split: 0.8413943312877885

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 20 split: 0.8532054948570542

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 21 split: 0.8454897297274008

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 22 split: 0.8230113565980401

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))
```

```
AUC Score for 23 split: 0.852983597940021
/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: F
ound `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".forma
t(alias))

AUC Score for 24 split: 0.8395965353920196

/usr/local/lib/python3.6/dist-packages/lightgbm/engine.py:118: UserWarning: Fou
nd `num_iterations` in params. Will use it instead of argument
  warnings.warn("Found `{}` in params. Will use it instead of argument".format
(alias))

AUC Score for 25 split: 0.8548007828867927
Final score: 0.8417761442656212
```

After several trials, the above model combination gave the best auc score

Catboost model prediction

In [150]:
```
default_status1 = Final_pred1
```

Votingclassifier(xgboost and catboost) prediction

In [151]:
```
default_status2 = Final_pred2
```

Votingclassifier(lightgbm, catboost and randomforest) prediction

In [152]:
```
default_status3 = Final_pred3
```

Weighted average of catboost model and votingclassifier(with xgboost and catboost)

In [153]:
```
Final_pred4 = Final_pred1*0.4 + Final_pred2*0.6
```

Weighted average of votingclassifier(with lightgbm, catboost and randomforest) and voting classifier(with xgboost and catboost)

In [154]:
```
Final_pred5 = Final_pred3*0.7 + Final_pred2*0.3
```

Weighted average of votingclassifier(with lightgbm, catboost and randomforest) and weighted average of catboost model and votingclassifier(with xgboost and catboost)

In [195]:
```
Final_pred6 = Final_pred3*0.7 + Final_pred4*0.3
```

In [196]:
```
default_status4 = Final_pred4
default_status5 = Final_pred5
default_status6 = Final_pred6
```

In [197]:
```
Test = Test.reset_index()
```

Predicted values for votingclassifier with lightgbm, catboost and randomforest.

In [198]:
```
predicted_values3 = pd.DataFrame({'Applicant_ID': Test['Applicant_ID'], 'default_
predicted_values3
```

Out[198]:

	Applicant_ID	default_status
0	Apcnt_1000032	0.422178
1	Apcnt_1000048	0.404793
2	Apcnt_1000052	0.495254
3	Apcnt_1000076	0.776757
4	Apcnt_1000080	0.229802
...	...	...
23995	Apcnt_999940	0.736293
23996	Apcnt_999956	0.356947
23997	Apcnt_999976	0.373381
23998	Apcnt_999984	0.615068
23999	Apcnt_999992	0.276300

24000 rows × 2 columns

Predicted values for weighted average of votingclassifier(with lightgbm, catboost and randomforest) and votingclassifier(with catboost and xgboost).

In [199]:
```
predicted_values5 = pd.DataFrame({'Applicant_ID': Test['Applicant_ID'], 'default_
predicted_values5
```

Out[199]:

	Applicant_ID	default_status
0	Apcnt_1000032	0.429166
1	Apcnt_1000048	0.417463
2	Apcnt_1000052	0.511428
3	Apcnt_1000076	0.792466
4	Apcnt_1000080	0.237686
...	...	...
23995	Apcnt_999940	0.754724
23996	Apcnt_999956	0.367176
23997	Apcnt_999976	0.376299
23998	Apcnt_999984	0.617652
23999	Apcnt_999992	0.277152

24000 rows × 2 columns

Predicted values for weighted average of votingclassifier(with lightgbm, catboost and randomforest) and the weighted average of catboost model and votingclassifier(with catboost and xgboost). This gave me my best auc score on the leaderboard.

In [200]:
```
predicted_values6 = pd.DataFrame({'Applicant_ID': Test['Applicant_ID'], 'default_
predicted_values6
```

Out[200]:

	Applicant_ID	default_status
0	Apcnt_1000032	0.411899
1	Apcnt_1000048	0.399594
2	Apcnt_1000052	0.497011
3	Apcnt_1000076	0.783711
4	Apcnt_1000080	0.227037
...	...	...
23995	Apcnt_999940	0.748793
23996	Apcnt_999956	0.354005
23997	Apcnt_999976	0.359800
23998	Apcnt_999984	0.606497
23999	Apcnt_999992	0.267104

24000 rows × 2 columns

```
In [201]: predicted_values3.to_csv('FinSub1.csv', index = False)
```

```
In [202]: predicted_values5.to_csv('FinSub2.csv', index = False)
```

The best and final Submission file

```
In [203]: predicted_values6.to_csv('FinSub3.csv', index = False)
```