In [2]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib inline
```

In [3]:
```
#Importing the data
data_train = pd.read_csv('Train.csv')
data_test = pd.read_csv('Test.csv')
```

In [4]:
```
data_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

5 rows × 52 columns

In [5]:
```
data_test.head()
```

Out[5]:

	Applicant_ID	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_f
0	Apcnt_1000032	3236.0	0.34875	10.2006	0.0000	0.0	418564.0	418
1	Apcnt_1000048	3284.0	1.27360	2.9606	9.0198	0.0	0.0	9858
2	Apcnt_1000052	NaN	0.27505	0.0600	0.0000	0.0	NaN	
3	Apcnt_1000076	3232.0	0.28505	2.8032	0.0000	0.0	0.0	473
4	Apcnt_1000080	3466.0	2.09545	0.8318	2.5182	0.0	19839.0	1150

5 rows × 51 columns

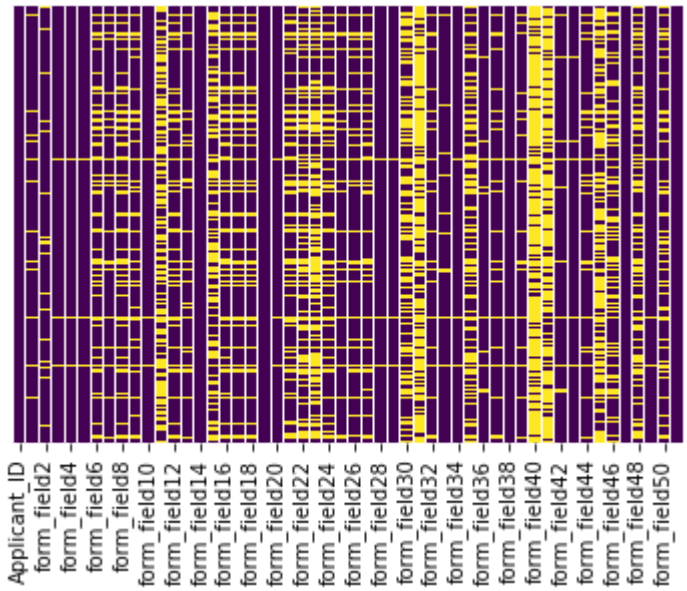
In [6]:
```
data_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56000 entries, 0 to 55999
Data columns (total 52 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Applicant_ID    56000 non-null  object 
 1   form_field1     53471 non-null  float64
 2   form_field2     52156 non-null  float64
 3   form_field3     55645 non-null  float64
 4   form_field4     55645 non-null  float64
 5   form_field5     55645 non-null  float64
 6   form_field6     42640 non-null  float64
 7   form_field7     50837 non-null  float64
 8   form_field8     42640 non-null  float64
 9   form_field9     47992 non-null  float64
 10  form_field10    55645 non-null  float64
 11  form_field11    24579 non-null  float64
 12  form_field12    46105 non-null  float64
 13  form_field13    50111 non-null  float64
 14  form_field14    56000 non-null  int64  
 15  form_field15    33525 non-null  float64
 16  form_field16    42964 non-null  float64
 17  form_field17    44849 non-null  float64
 18  form_field18    45598 non-null  float64
 19  form_field19    55996 non-null  float64
 20  form_field20    55645 non-null  float64
 21  form_field21    40146 non-null  float64
 22  form_field22    35600 non-null  float64
 23  form_field23    27877 non-null  float64
 24  form_field24    42703 non-null  float64
 25  form_field25    50550 non-null  float64
 26  form_field26    48562 non-null  float64
 27  form_field27    46701 non-null  float64
 28  form_field28    55645 non-null  float64
 29  form_field29    55645 non-null  float64
 30  form_field30    30491 non-null  float64
 31  form_field31    16592 non-null  float64
 32  form_field32    50550 non-null  float64
 33  form_field33    54744 non-null  float64
 34  form_field34    55645 non-null  float64
 35  form_field35    32852 non-null  float64
 36  form_field36    54005 non-null  float64
 37  form_field37    50550 non-null  float64
 38  form_field38    55645 non-null  float64
 39  form_field39    51789 non-null  float64
 40  form_field40    12271 non-null  float64
 41  form_field41    17771 non-null  float64
 42  form_field42    54677 non-null  float64
 43  form_field43    55432 non-null  float64
 44  form_field44    50617 non-null  float64
 45  form_field45    24683 non-null  float64
 46  form_field46    40096 non-null  float64
 47  form_field47    56000 non-null  object 
 48  form_field48    35111 non-null  float64
 49  form_field49    55645 non-null  float64
```

```
 50  form_field50    44944 non-null  float64
 51  default_status  56000 non-null  object
dtypes: float64(48), int64(1), object(3)
memory usage: 22.2+ MB
```

In [7]:
```
#Checking for missing values
sns.heatmap(data_train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x1f38530c348>



In [8]:
```
#Checking for null values
data_train.isnull().sum()
```

Out[8]:
```
Applicant_ID         0
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
form_field21     15854
form_field22     20400
form_field23     28123
form_field24     13297
form_field25      5450
form_field26      7438
form_field27      9299
form_field28       355
form_field29       355
form_field30     25509
form_field31     39408
form_field32      5450
form_field33      1256
form_field34       355
form_field35     23148
form_field36      1995
form_field37      5450
form_field38       355
form_field39      4211
form_field40     43729
form_field41     38229
form_field42      1323
form_field43       568
form_field44      5383
form_field45     31317
form_field46     15904
form_field47         0
form_field48     20889
form_field49       355
form_field50     11056
default_status       0
dtype: int64
```

In [9]:
```
from sklearn.preprocessing import LabelEncoder
```

In [10]:
```
#Encoding to a numeric data
label_encoder = LabelEncoder()
data_train['form_field47'] = label_encoder.fit_transform(data_train['form_field47
```

In [11]:
```
data_train
```

Out[11]:

	Applicant_ID	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	fc
0	Apcnt_1000000	3436.0	0.28505	1.6560	0.0000	0.000	0.0	1
1	Apcnt_1000004	3456.0	0.67400	0.2342	0.0000	0.000	0.0	
2	Apcnt_1000008	3276.0	0.53845	3.1510	0.0000	6.282	NaN	
3	Apcnt_1000012	3372.0	0.17005	0.5050	0.0000	0.000	192166.0	
4	Apcnt_1000016	3370.0	0.77270	1.1010	0.0000	0.000	1556.0	
...	...	...	...	...	...	...	...	
55995	Apcnt_999968	3740.0	0.01730	0.0000	0.0000	0.000	770998.0	
55996	Apcnt_999972	3360.0	2.01145	0.6252	0.0000	0.000	NaN	
55997	Apcnt_999980	3500.0	0.76640	0.0000	0.0000	0.000	118645.0	
55998	Apcnt_999988	3280.0	0.05235	2.0916	2.2212	0.000	NaN	
55999	Apcnt_999996	3522.0	0.46930	0.0000	0.0000	0.000	98806.0	

56000 rows × 52 columns

In [12]:
```
data_train.columns
```

Out[12]:
```
Index(['Applicant_ID', '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_field47',
       'form_field48', 'form_field49', 'form_field50', 'default_status'],
      dtype='object')
```

In [13]:
```
#Columns to be used while training
feature_columns = ['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_field47',
       'form_field50', 'default_status']
```

In [14]:
```
data_train.drop('Applicant_ID',axis=1,inplace=True)
```

In [15]:
```
#Replacing missing
for column in data_train.columns:
    data_train_mean = data_train[column].mean()
    data_train[column].fillna(data_train_mean, inplace = True)
    print(data_train.isnull().sum())
```

```
form_field15      22475
form_field16      13036
form_field17      11151
form_field18      10402
form_field19          4
form_field20        355
form_field21      15854
form_field22      20400
form_field23      28123
form_field24      13297
form_field25       5450
form_field26       7438
form_field27       9299
form_field28        355
form_field29        355
form_field30      25509
form_field31      39408
form_field32       5450
form_field33       1256
form_field34        355
```

In [16]:
```
data_train
```

Out[16]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_field7	
0	3436.0	0.28505	1.6560	0.0000	0.000	0.000000	10689720.0	2.
1	3456.0	0.67400	0.2342	0.0000	0.000	0.000000	898979.0	4.
2	3276.0	0.53845	3.1510	0.0000	6.282	624447.924437	956940.0	2.
3	3372.0	0.17005	0.5050	0.0000	0.000	192166.000000	3044703.0	3.
4	3370.0	0.77270	1.1010	0.0000	0.000	1556.000000	214728.0	2.
...	...	...	...	...	...	...	...	
55995	3740.0	0.01730	0.0000	0.0000	0.000	770998.000000	9637475.0	4.
55996	3360.0	2.01145	0.6252	0.0000	0.000	624447.924437	927765.0	2.
55997	3500.0	0.76640	0.0000	0.0000	0.000	118645.000000	3662435.0	3.
55998	3280.0	0.05235	2.0916	2.2212	0.000	624447.924437	3458599.0	2.
55999	3522.0	0.46930	0.0000	0.0000	0.000	98806.000000	2053920.0	5.

56000 rows × 51 columns

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
data_train['default_status'].head(3)
```

Out[18]:
```
0     no
1     no
2    yes
Name: default_status, dtype: object
```

In [19]:
```
#Encoding the prediction data
target_encoder = LabelEncoder().fit(data_train['default_status'])
data_train['default_status'] = target_encoder.transform(data_train['default_statu
```

In [20]:
```
data_train['default_status'].head(5)
```

Out[20]:
```
0    0
1    0
2    1
3    0
4    0
Name: default_status, dtype: int32
```

In [21]:
```
#Features
feature_columns = ['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_field47', 'form_field48',
       'form_field49', 'form_field50']
```

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split( data_train[feature_columns],
```

In [23]:
```
from sklearn.linear_model import LogisticRegression
```

In [24]:
```
#Using Logistics Regression
logreg = LogisticRegression()
logreg.fit(X_train,y_train)
```

```
C:\Users\User\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:94
0: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-
learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on)
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[24]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [25]:
```
from sklearn.metrics import accuracy_score
```

In [26]:
```
predictions = logreg.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.7582142857142857
```

In [ ]:

In [27]:
```
#Importing other models
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
```

In [28]:
```
gb_model = GradientBoostingClassifier()
svc_model = LinearSVC()
rf_model = RandomForestClassifier()
ada_model = AdaBoostClassifier()
```

In [29]:
```
#Using GradientBoostClassifier
gb_model = gb_model.fit(X_train, y_train)
gb_pred = gb_model.predict(X_test)
accuracy = accuracy_score(y_test, gb_pred)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.8078571428571428
```

In [30]:
```
#Using RandomForestClassifier
rf_model = rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
accuracy = accuracy_score(y_test, rf_pred)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.807797619047619
```

In [31]:
```
#Testing AdaBoostClassifier
ada_model = AdaBoostClassifier(learning_rate=0.6)
ada_model = ada_model.fit(X_train, y_train)
ada_pred = ada_model.predict(X_test)
accuracy = accuracy_score(y_test, ada_pred)
print("Accuracy: ", accuracy)
```

```
Accuracy:  0.8003571428571429
```

PREDICTING TEST

In [32]:
```
data_test.head(5)
```

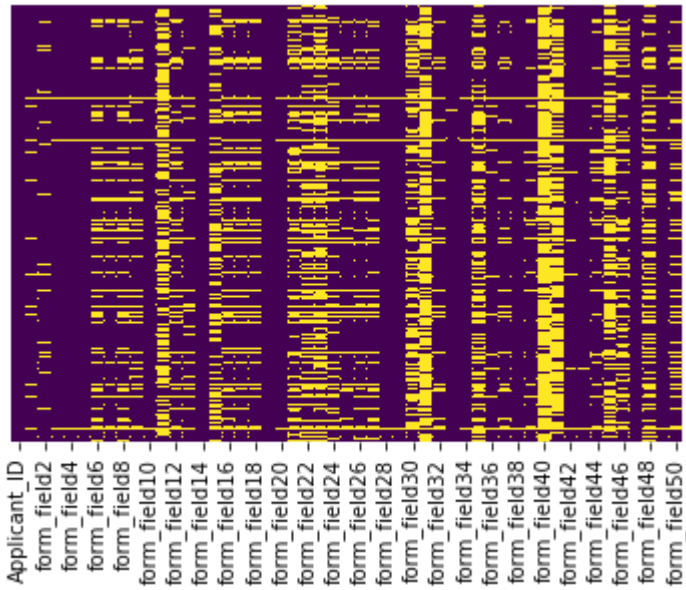
Out[32]:

	Applicant_ID	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_f
0	Apcnt_1000032	3236.0	0.34875	10.2006	0.0000	0.0	418564.0	418
1	Apcnt_1000048	3284.0	1.27360	2.9606	9.0198	0.0	0.0	9858
2	Apcnt_1000052	NaN	0.27505	0.0600	0.0000	0.0	NaN	
3	Apcnt_1000076	3232.0	0.28505	2.8032	0.0000	0.0	0.0	473
4	Apcnt_1000080	3466.0	2.09545	0.8318	2.5182	0.0	19839.0	1150

5 rows × 51 columns

In [33]:
```
sns.heatmap(data_test.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f385bd1288>
```



In [34]:
```
data_test.isnull().sum()
```

Out[34]:
```
Applicant_ID        0
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
form_field21     6707
form_field22     8724
form_field23    12125
form_field24     5605
form_field25     2256
form_field26     3172
form_field27     3910
form_field28      147
form_field29      147
form_field30    10908
form_field31    16810
form_field32     2256
form_field33      495
form_field34      147
form_field35     9866
form_field36      903
form_field37     2256
form_field38      147
form_field39     1829
form_field40    18828
form_field41    16349
form_field42      578
form_field43      250
form_field44     2362
form_field45    13538
form_field46     6885
form_field47        0
form_field48     8922
form_field49      146
form_field50     4797
dtype: int64
```

In [35]:
```
label_encoder = LabelEncoder()
data_test['form_field47'] = label_encoder.fit_transform(data_test['form_field47'])
```

In [36]:
```
data_test.drop('Applicant_ID',axis=1,inplace=True)
```

In [38]:
```
for column in data_test.columns:
    data_test_mean = data_test[column].mean()
    data_test[column].fillna(data_train_mean, inplace = True)
    print(data_test.isnull().sum())
```

```
form_field1     0
form_field2     0
form_field3     0
form_field4     0
form_field5     0
form_field6     0
form_field7     0
form_field8     0
form_field9     0
form_field10    0
form_field11    0
form_field12    0
form_field13    0
form_field14    0
form_field15    0
form_field16    0
form_field17    0
form_field18    0
form_field19    0
```

In [39]:
```
data_test.head(3)
```

Out[39]:

	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_field7	
0	3236.000000	0.34875	10.2006	0.0000	0.0	418564.000000	4.185640e+05	4
1	3284.000000	1.27360	2.9606	9.0198	0.0	0.000000	9.858816e+06	
2	600586.172883	0.27505	0.0600	0.0000	0.0	600586.172883	6.005862e+05	6

3 rows × 50 columns

In [40]:
```
label_encoder = LabelEncoder()
data_test['form_field47'] = label_encoder.fit_transform(data_test['form_field47'])
```

In [41]:
```
data_train.columns
```

Out[41]:
```
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_field47', 'form_field48',
       'form_field49', 'form_field50', 'default_status'],
      dtype='object')
```

In [42]:
```
data_test.columns
```

Out[42]:
```
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_field47', 'form_field48',
       'form_field49', 'form_field50'],
      dtype='object')
```

In [43]:
```
#Features
feature_columns = ['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_field47', 'form_field48',
       'form_field49', 'form_field50']
```

In [44]:
```
test_pred = rf_model.predict(data_test[feature_columns])
```

In [45]:
```
#Our Test Prediction
set(test_pred)
```

Out[45]:
```
{0, 1}
```

In [46]:
```
Submission = pd.read_csv('SampleSubmission.csv')
```

In [47]:
```
Submission.head()
```

Out[47]:

	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 [ ]:
```

```

In [48]:
```
New_Test = pd.read_csv('Test.csv')
```

In [49]:
```
New_Test.head()
```

Out[49]:

	Applicant_ID	form_field1	form_field2	form_field3	form_field4	form_field5	form_field6	form_f
0	Apcnt_1000032	3236.0	0.34875	10.2006	0.0000	0.0	418564.0	418
1	Apcnt_1000048	3284.0	1.27360	2.9606	9.0198	0.0	0.0	9858
2	Apcnt_1000052	NaN	0.27505	0.0600	0.0000	0.0	NaN	
3	Apcnt_1000076	3232.0	0.28505	2.8032	0.0000	0.0	0.0	473
4	Apcnt_1000080	3466.0	2.09545	0.8318	2.5182	0.0	19839.0	1150

5 rows × 51 columns

In [50]:
```
print(test_pred)
```

```
[1 0 0 ... 0 1 0]
```

In [51]:
```
my_submission = pd.DataFrame({"Applicant_ID": New_Test["Applicant_ID"], 'default_
my_submission.head()
```

Out[51]:

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

In [52]:
```
my_submission.to_csv("My_Submission.csv", index=False)
```

In [ ]:
```

```