In [50]:

```
import pandas as pd
import matplotlib.pyplot as ply
import seaborn as sns
import numpy as np
import scipy.stats as st
import statsmodels.formula.api as smf
```

In [51]:

```
import pandas as pd
import matplotlib.pyplot as ply
import seaborn as sns
import numpy as np
import scipy.stats as st
# Load libraries
import pandas as pd
#import numpy
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model
import statsmodels.formula.api as smf
from sklearn import linear_model
from patsy.builtins import Q
from sklearn.linear_model import RidgeCV
```

In [52]:

```
df = pd.read_csv("LoanStats3b.csv",low_memory =False)
```

In [53]:

```
#df.head(10)
```

In [54]:

```
print('Number of missing values per column:')
countMissing = df.isnull().sum()
print(countMissing)
```

```
Number of missing values per column:
```

```
id                                      188181
member_id                               188181
loan_amnt                                    0
funded_amnt                                  0
funded_amnt_inv                              0
term                                         0
int_rate                                     0
installment                                  0
grade                                        0
sub_grade                                    0
emp_title                                11737
emp_length                                7887
home_ownership                               0
annual_inc                                   0
verification_status                          0
issue_d                                      0
loan_status                                  0
pymnt_plan                                   0
url                                     188181
desc                                    106703
purpose                                      0
title                                        7
zip_code                                     0
addr_state                                   0
dti                                          0
delinq_2yrs                                  0
earliest_cr_line                             0
inq_last_6mths                               0
mths_since_last_delinq                  107573
mths_since_last_record                  170707
                                           ...
sec_app_mort_acc                        188181
sec_app_open_acc                        188181
sec_app_revol_util                      188181
sec_app_open_act_il                     188181
sec_app_num_rev_accts                   188181
sec_app_chargeoff_within_12_mths        188181
sec_app_collections_12_mths_ex_med      188181
sec_app_mths_since_last_major_derog     188181
hardship_flag                                0
hardship_type                           188101
hardship_reason                         188101
hardship_status                         188101
deferral_term                           188101
hardship_amount                         188101
```

```
hardship_start_date                                188101
hardship_end_date                                  188101
payment_plan_start_date                            188101
hardship_length                                    188101
hardship_dpd                                       188101
hardship_loan_status                               188101
orig_projected_additional_accrued_interest         188112
hardship_payoff_balance_amount                     188101
hardship_last_payment_amount                       188101
debt_settlement_flag                                    0
debt_settlement_flag_date                          186103
settlement_status                                  186103
settlement_date                                    186103
settlement_amount                                  186103
settlement_percentage                              186103
settlement_term                                    186103
Length: 144, dtype: int64
```

In [55]:

```
# Drop the columns where all elements are missing
#df = df.dropna(axis=0,how='any')
df = df.dropna(axis=1,how='all')
```

In [56]:

```
# Columns that are being dropped
df=df.drop(['hardship_dpd','hardship_loan_status','emp_title','hardship_type','hardship_reason','hardship_status','deferral_term','hardship_amount'],axis=1)
df=df.drop(['settlement_status','settlement_date','settlement_amount','settlement_term', 'policy_code', 'acc_now_delinq', 'num_tl_30dpd'],axis=1)
df=df.drop(['hardship_start_date','hardship_end_date','orig_projected_additional_accrued_interest','hardship_payoff_balance_amount','debt_settlement_flag_date'],axis=1)
df=df.drop(['total_bal_ex_mort', 'num_sats', 'tot_cur_bal', 'tax_liens', 'zip_code','addr_state','title', 'num_tl_90g_dpd_24m'], axis=1)
df=df.drop(['payment_plan_start_date','hardship_length','hardship_last_payment_amount','settlement_percentage', 'collections_12_mths_ex_med'],axis=1)
df=df.drop(['mths_since_last_record', 'mths_since_recent_bc_dlq', 'mths_since_recent_revol_delinq', 'mths_since_last_delinq' ,'mths_since_last_major_derog'],axis=1)
df=df.drop(['hardship_flag', 'debt_settlement_flag', 'num_tl_120dpd_2m', 'chargeoff_within_12_mths', 'delinq_amnt', 'application_type', 'emp_length'],axis=1)
df=df.drop(['out_prncp', 'out_prncp_inv'], axis=1)
```

In [57]:

```
print('Number of missing values per column:')
countMissing = df.isnull().sum()
print(countMissing)
```

```
Number of missing values per column:
loan_amnt                       0
funded_amnt                     0
funded_amnt_inv                 0
term                            0
int_rate                        0
installment                     0
grade                           0
sub_grade                       0
home_ownership                  0
annual_inc                      0
verification_status             0
```

```
issue_d                       0
loan_status                   0
pymnt_plan                    0
desc                     106703
purpose                       0
dti                           0
delinq_2yrs                   0
earliest_cr_line              0
inq_last_6mths                0
open_acc                      0
pub_rec                       0
revol_bal                     0
revol_util                  125
total_acc                     0
initial_list_status           0
total_pymnt                   0
total_pymnt_inv               0
total_rec_prncp               0
total_rec_int                 0
                            ...
last_credit_pull_d           10
tot_coll_amt              27741
total_rev_hi_lim          27741
acc_open_past_24mths       7495
avg_cur_bal               27747
bc_open_to_buy             9025
bc_util                    9112
mo_sin_old_il_acct        33872
mo_sin_old_rev_tl_op      27742
mo_sin_rcnt_rev_tl_op     27742
mo_sin_rcnt_tl            27741
mort_acc                   7495
mths_since_recent_bc       8828
mths_since_recent_inq     27868
num_accts_ever_120_pd     27741
num_actv_bc_tl            27741
num_actv_rev_tl           27741
num_bc_sats               16055
num_bc_tl                 27741
num_il_tl                 27741
num_op_rev_tl             27741
num_rev_accts             27741
num_rev_tl_bal_gt_0       27741
num_tl_op_past_12m        27741
pct_tl_nvr_dlq            27894
```

```
percent_bc_gt_75                9028
pub_rec_bankruptcies               0
tot_hi_cred_lim                27741
total_bc_limit                  7495
total_il_high_credit_limit     27741
Length: 65, dtype: int64
```

In [58]:

```
# df = df.drop(df[df['revol_util']==0].index)
# df.drop(df.index[df['last_credit_pull_d'] == 0], inplace = Tru
e)
```

In [59]:

```
df['int_rate'] = df['int_rate'].str.rstrip('%')
df['int_rate'] = df['int_rate'].astype('float64')
```

In [60]:

```
numericalList = []
nonNumList = []
for column in df.columns:
    if df[column].dtypes == 'int64' or df[column].dtypes == 'flo
at64':
        numericalList.append(column)
    else:
        nonNumList.append(column)
```

In [ ]:

```

```

In [61]:

```
numDF = df[numericalList]
nonnumDF = df[nonNumList]
```

In [62]:

```
nrow = len(numDF['int_rate'])
# print('Number of missing values per column:')
num_countMissing = numDF.isnull().sum()
# print(num_countMissing)
```

In [63]:

```
# Estimate the mean for columns that are missing less than 20% o
f the observations
for coln in numDF.columns:
    if num_countMissing[coln] != 0:
        if num_countMissing[coln]/nrow < 0.20:
            numDF[coln] = numDF[coln].fillna(value=numDF[coln].m
ean())
```

```
/Users/michellebaginski/anaconda3/lib/python3.7/site
-packages/ipykernel_launcher.py:5: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice fro
m a DataFrame.
Try using .loc[row_indexer,col_indexer] = value inst
ead

See the caveats in the documentation: http://pandas.
pydata.org/pandas-docs/stable/indexing.html#indexing
-view-versus-copy
  """
```

In [64]:

```
# remove all the observations with NAs in them
df = df.dropna(axis=0,how='any')

print('Number of missing values per column:')
countMissing = df.isnull().sum()
print(countMissing)
```

```
Number of missing values per column:
loan_amnt                       0
funded_amnt                     0
funded_amnt_inv                 0
term                            0
```

```
int_rate                    0
installment                 0
grade                       0
sub_grade                   0
home_ownership              0
annual_inc                  0
verification_status         0
issue_d                     0
loan_status                 0
pymnt_plan                  0
desc                        0
purpose                     0
dti                         0
delinq_2yrs                 0
earliest_cr_line            0
inq_last_6mths              0
open_acc                    0
pub_rec                     0
revol_bal                   0
revol_util                  0
total_acc                   0
initial_list_status         0
total_pymnt                 0
total_pymnt_inv             0
total_rec_prncp             0
total_rec_int               0
                           ..
last_credit_pull_d          0
tot_coll_amt                0
total_rev_hi_lim            0
acc_open_past_24mths        0
avg_cur_bal                 0
bc_open_to_buy              0
bc_util                     0
mo_sin_old_il_acct          0
mo_sin_old_rev_tl_op        0
mo_sin_rcnt_rev_tl_op       0
mo_sin_rcnt_tl              0
mort_acc                    0
mths_since_recent_bc        0
mths_since_recent_inq       0
num_accts_ever_120_pd       0
num_actv_bc_tl              0
num_actv_rev_tl             0
num_bc_sats                 0
```

```
num_bc_tl                     0
num_il_tl                     0
num_op_rev_tl                 0
num_rev_accts                 0
num_rev_tl_bal_gt_0           0
num_tl_op_past_12m            0
pct_tl_nvr_dlq                0
percent_bc_gt_75              0
pub_rec_bankruptcies          0
tot_hi_cred_lim               0
total_bc_limit                0
total_il_high_credit_limit    0
Length: 65, dtype: int64
```

In [65]:

```
# Removing variables on the basis of correlation matrix
corr = numDF.corr()
print(corr)
columns = np.full((corr.shape[0],), True, dtype=bool)
for i in range(corr.shape[0]):
    for j in range(i+1, corr.shape[0]):
        if corr.iloc[i,j] >= 0.8:
            if columns[j]:
                columns[j] = False
selected_columns = numDF.columns[columns]
```

```
                          loan_amnt  funded_amnt
funded_amnt_inv  int_rate  \
loan_amnt                  1.000000     0.999799
0.999663  0.182654
funded_amnt                0.999799     1.000000
0.999874  0.182485
funded_amnt_inv            0.999663     0.999874
1.000000  0.182933
int_rate                   0.182654     0.182485
0.182933  1.000000
installment                0.955011     0.955254
0.955211  0.165173
annual_inc                 0.368164     0.368151
0.368084 -0.026026
dti                        0.044557     0.044572
0.044746  0.147471
delinq_2yrs                0.011184     0.011214
0.011391  0.097230
```

```
inq_last_6mths              0.019741     0.019703
0.020091  0.241345
open_acc                    0.191571     0.191614
0.191719  0.017359
pub_rec                    -0.073588    -0.073537
-0.073373  0.056575
revol_bal                   0.320262     0.320284
0.320259 -0.003159
total_acc                   0.238362     0.238363
0.238460 -0.019417
total_pymnt                 0.891293     0.891500
0.891614  0.203856
total_pymnt_inv             0.891239     0.891456
0.891635  0.204200
total_rec_prncp             0.844267     0.844501
0.844561  0.041107
total_rec_int               0.696224     0.696306
0.696515  0.480275
total_rec_late_fee          0.079565     0.079627
0.079690  0.079482
recoveries                  0.190052     0.190055
0.190051  0.181272
collection_recovery_fee     0.156731     0.156767
0.156717  0.139377
last_pymnt_amnt             0.432104     0.432127
0.432178  0.126608
tot_coll_amt               -0.019016    -0.019020
-0.019030  0.010068
total_rev_hi_lim            0.237312     0.237367
0.237346 -0.143858
acc_open_past_24mths        0.003601     0.003557
0.003667  0.150894
avg_cur_bal                 0.216832     0.216889
0.216840 -0.127608
bc_open_to_buy              0.175445     0.175497
0.175417 -0.340119
bc_util                     0.040698     0.040695
0.040768  0.374959
mo_sin_old_il_acct          0.135317     0.135350
0.135358 -0.031711
mo_sin_old_rev_tl_op        0.173151     0.173190
0.173189 -0.109867
mo_sin_rcnt_rev_tl_op       0.043687     0.043701
0.043624 -0.100733
mo_sin_rcnt_tl              0.008197     0.008203
```

```
0.008097 -0.124643
mort_acc                      0.234556     0.234613
0.234613 -0.096496
mths_since_recent_bc          0.035696     0.035692
0.035657 -0.047071
mths_since_recent_inq        -0.000066    -0.000041
-0.000482 -0.209414
num_accts_ever_120_pd        -0.049013    -0.049041
-0.049017  0.073939
num_actv_bc_tl                0.153285     0.153323
0.153374  0.034850
num_actv_rev_tl               0.125833     0.125862
0.125937  0.124939
num_bc_sats                   0.183156     0.183154
0.183168 -0.061437
num_bc_tl                     0.173745     0.173784
0.173779 -0.092251
num_il_tl                     0.089742     0.089753
0.089812  0.038724
num_op_rev_tl                 0.152853     0.152881
0.152924 -0.009553
num_rev_accts                 0.172846     0.172877
0.172889 -0.060861
num_rev_tl_bal_gt_0           0.125901     0.125930
0.126005  0.125288
num_tl_op_past_12m           -0.008921    -0.008948
-0.008826  0.184984
pct_tl_nvr_dlq                0.072673     0.072692
0.072627 -0.115032
percent_bc_gt_75              0.007203     0.007233
0.007298  0.353748
pub_rec_bankruptcies         -0.094852    -0.094798
-0.094606  0.048524
tot_hi_cred_lim               0.306396     0.306465
0.306440 -0.155279
total_bc_limit                0.358044     0.358130
0.358048 -0.261619
total_il_high_credit_limit    0.173811     0.173821
0.173894  0.024808

                           installment  annual_inc
dti  delinq_2yrs  \
loan_amnt                    0.955011    0.368164
0.044557     0.011184
```

```
funded_amnt                  0.955254    0.368151
0.044572     0.011214
funded_amnt_inv              0.955211    0.368084
0.044746     0.011391
int_rate                     0.165173   -0.026026
0.147471     0.097230
installment                  1.000000    0.367857
0.039438     0.022610
annual_inc                   0.367857    1.000000
-0.196529     0.069232
dti                          0.039438   -0.196529
1.000000    -0.009784
delinq_2yrs                  0.022610    0.069232
-0.009784     1.000000
inq_last_6mths               0.039678    0.085247
0.011601     0.025841
open_acc                     0.187049    0.159963
0.302366     0.056357
pub_rec                     -0.065317   -0.023327
-0.053917    -0.022420
revol_bal                    0.310904    0.341650
0.145847    -0.024285
total_acc                    0.221266    0.238645
0.230711     0.134704
total_pymnt                  0.838972    0.336200
0.042459     0.020337
total_pymnt_inv              0.839014    0.336202
0.042551     0.020470
total_rec_prncp              0.821796    0.347716
0.003891     0.005934
total_rec_int                0.596202    0.203455
0.108085     0.044175
total_rec_late_fee           0.078131    0.032474
0.011868     0.030405
recoveries                   0.162123    0.034279
0.049467     0.015664
collection_recovery_fee      0.129182    0.030331
0.046153     0.015142
last_pymnt_amnt              0.385657    0.191088
-0.028060     0.001715
tot_coll_amt                -0.017793   -0.001675
-0.013457     0.004088
total_rev_hi_lim             0.217850    0.263770
0.048801    -0.023461
acc_open_past_24mths         0.013906    0.049138
```

0.157777 -0.058441
avg_cur_bal 0.186223 0.374005
-0.119971 0.063175
bc_open_to_buy 0.138873 0.166127
-0.091126 -0.032432
bc_util 0.073825 -0.019572
0.206253 -0.017904
mo_sin_old_il_acct 0.117916 0.134886
0.038983 0.079093
mo_sin_old_rev_tl_op 0.151669 0.150621
0.035009 0.094795
mo_sin_rcnt_rev_tl_op 0.029200 0.038323
-0.023834 0.039344
mo_sin_rcnt_tl -0.001376 -0.028659
-0.093251 0.023639
mort_acc 0.197396 0.274060
-0.042512 0.105114
mths_since_recent_bc 0.024574 0.043251
-0.002119 0.069526
mths_since_recent_inq -0.018969 -0.050274
0.000968 -0.018265
num_accts_ever_120_pd -0.038043 0.028157
-0.057088 0.216533
num_actv_bc_tl 0.161251 0.074031
0.147802 -0.058097
num_actv_rev_tl 0.139411 0.046627
0.231110 -0.025010
num_bc_sats 0.181600 0.101672
0.093352 -0.047838
num_bc_tl 0.167666 0.131620
0.064316 0.041219
num_il_tl 0.080364 0.131219
0.239445 0.083041
num_op_rev_tl 0.153466 0.068402
0.157211 0.006416
num_rev_accts 0.166111 0.120787
0.114310 0.082627
num_rev_tl_bal_gt_0 0.139449 0.046725
0.231599 -0.024370
num_tl_op_past_12m 0.008082 0.050567
0.097471 -0.036260
pct_tl_nvr_dlq 0.053564 -0.023768
0.083720 -0.436693
percent_bc_gt_75 0.036409 -0.038702

```
0.188117    -0.021144
pub_rec_bankruptcies          -0.088497   -0.054342
-0.054327    -0.038079
tot_hi_cred_lim                0.270641    0.481292
-0.004814     0.080387
total_bc_limit                 0.323503    0.288585
0.031969    -0.059412
total_il_high_credit_limit     0.164982    0.292326
0.322581     0.068556

                            inq_last_6mths  open_acc
...  num_op_rev_tl  \
loan_amnt                         0.019741  0.191571
...       0.152853
funded_amnt                       0.019703  0.191614
...       0.152881
funded_amnt_inv                   0.020091  0.191719
...       0.152924
int_rate                          0.241345  0.017359
...      -0.009553
installment                       0.039678  0.187049
...       0.153466
annual_inc                        0.085247  0.159963
...       0.068402
dti                               0.011601  0.302366
...       0.157211
delinq_2yrs                       0.025841  0.056357
...       0.006416
inq_last_6mths                    1.000000  0.125784
...       0.089312
open_acc                          0.125784  1.000000
...       0.760595
pub_rec                           0.010963 -0.033834
...      -0.018137
revol_bal                         0.008154  0.217770
...       0.214840
total_acc                         0.154447  0.666391
...       0.459027
total_pymnt                       0.007352  0.167302
...       0.131121
total_pymnt_inv                   0.007723  0.167393
...       0.131154
total_rec_prncp                  -0.016885  0.159681
...       0.127439
total_rec_int                     0.051162  0.125943
```

```
...       0.093824
total_rec_late_fee                0.013419  0.010754
...      -0.002819
recoveries                        0.043446  0.043598
...       0.032373
collection_recovery_fee           0.026198  0.041811
...       0.032996
last_pymnt_amnt                   0.055317  0.082393
...       0.048531
tot_coll_amt                      0.011059  0.005173
...       0.002767
total_rev_hi_lim                  0.030182  0.245941
...       0.299210
acc_open_past_24mths              0.219278  0.436268
...       0.328295
avg_cur_bal                       0.048265 -0.081989
...      -0.190952
bc_open_to_buy                    0.037227  0.237378
...       0.280663
bc_util                          -0.081087 -0.086000
...      -0.124842
mo_sin_old_il_acct                0.013917  0.110979
...       0.055859
mo_sin_old_rev_tl_op             -0.002869  0.132323
...       0.193083
mo_sin_rcnt_rev_tl_op            -0.137293 -0.211942
...      -0.277887
mo_sin_rcnt_tl                   -0.202117 -0.212953
...      -0.186244
mort_acc                          0.097495  0.128045
...       0.056109
mths_since_recent_bc             -0.091008 -0.187125
...      -0.223272
mths_since_recent_inq            -0.640108 -0.085326
...      -0.064062
num_accts_ever_120_pd             0.048480  0.007914
...      -0.019155
num_actv_bc_tl                    0.016336  0.471667
...       0.636410
num_actv_rev_tl                   0.047663  0.597288
...       0.794731
num_bc_sats                       0.052308  0.582938
...       0.725595
num_bc_tl                         0.086334  0.430927
```

```
...       0.563712
num_il_tl                          0.089378  0.354235
...      -0.008681
num_op_rev_tl                      0.089312  0.760595
...       1.000000
num_rev_accts                      0.109858  0.568211
...       0.730405
num_rev_tl_bal_gt_0                0.048051  0.598248
...       0.795936
num_tl_op_past_12m                 0.244767  0.297025
...       0.257699
pct_tl_nvr_dlq                    -0.017770  0.077882
...       0.106137
percent_bc_gt_75                  -0.078196 -0.092029
...      -0.128804
pub_rec_bankruptcies               0.004385 -0.048503
...      -0.026013
tot_hi_cred_lim                    0.097522  0.255636
...       0.111189
total_bc_limit                     0.009950  0.301572
...       0.340577
total_il_high_credit_limit         0.094896  0.334169
...       0.002144

                           num_rev_accts  num_rev_t
l_bal_gt_0  \
loan_amnt                       0.172846
0.125901
funded_amnt                     0.172877
0.125930
funded_amnt_inv                 0.172889
0.126005
int_rate                       -0.060861
0.125288
installment                     0.166111
0.139449
annual_inc                      0.120787
0.046725
dti                             0.114310
0.231599
delinq_2yrs                     0.082627
-0.024370
inq_last_6mths                  0.109858
0.048051
open_acc                        0.568211
```

0.598248
pub_rec -0.012743
-0.017076
revol_bal 0.202645
0.235843
total_acc 0.706895
0.306916
total_pymnt 0.144342
0.120143
total_pymnt_inv 0.144347
0.120207
total_rec_prncp 0.153070
0.092193
total_rec_int 0.075496
0.139935
total_rec_late_fee -0.005876
0.007195
recoveries 0.027268
0.043519
collection_recovery_fee 0.028624
0.042869
last_pymnt_amnt 0.095749
0.007208
tot_coll_amt 0.022182
-0.013133
total_rev_hi_lim 0.271473
0.197993
acc_open_past_24mths 0.283202
0.231467
avg_cur_bal -0.045717
-0.182119
bc_open_to_buy 0.260654
0.061764
bc_util -0.132351
0.121878
mo_sin_old_il_acct 0.151079
0.055949
mo_sin_old_rev_tl_op 0.340432
0.136856
mo_sin_rcnt_rev_tl_op -0.224745
-0.217148
mo_sin_rcnt_tl -0.162482
-0.136792
mort_acc 0.204279

```
0.015141
mths_since_recent_bc            -0.167579
-0.178550
mths_since_recent_inq           -0.078780
-0.037753
num_accts_ever_120_pd            0.107531
-0.029712
num_actv_bc_tl                   0.392016
0.790457
num_actv_rev_tl                  0.509936
0.998491
num_bc_sats                      0.498947
0.628752
num_bc_tl                        0.850541
0.416351
num_il_tl                        0.081672
-0.021621
num_op_rev_tl                    0.730405
0.795936
num_rev_accts                    1.000000
0.510600
num_rev_tl_bal_gt_0              0.510600
1.000000
num_tl_op_past_12m               0.226864
0.178580
pct_tl_nvr_dlq                   0.004681
0.097005
percent_bc_gt_75                -0.132262
0.098695
pub_rec_bankruptcies            -0.014779
-0.023540
tot_hi_cred_lim                  0.180928
0.059625
total_bc_limit                   0.313148
0.196284
total_il_high_credit_limit       0.043024
-0.012111

                          num_tl_op_past_12m  pct_
tl_nvr_dlq  \
loan_amnt                         -0.008921
0.072673
funded_amnt                       -0.008948
0.072692
funded_amnt_inv                   -0.008826
```

0.072627
int_rate 0.184984
-0.115032
installment 0.008082
0.053564
annual_inc 0.050567
-0.023768
dti 0.097471
0.083720
delinq_2yrs -0.036260
-0.436693
inq_last_6mths 0.244767
-0.017770
open_acc 0.297025
0.077882
pub_rec 0.000880
0.006628
revol_bal -0.024168
0.116028
total_acc 0.262375
-0.016746
total_pymnt -0.026924
0.049632
total_pymnt_inv -0.026818
0.049573
total_rec_prncp -0.046609
0.063754
total_rec_int 0.014470
0.001775
total_rec_late_fee 0.009647
-0.023673
recoveries 0.050521
0.002297
collection_recovery_fee 0.035131
0.002102
last_pymnt_amnt 0.044297
0.040964
tot_coll_amt 0.011436
-0.062637
total_rev_hi_lim 0.036456
0.125221
acc_open_past_24mths 0.665080
0.047643
avg_cur_bal -0.018951

-0.040238
bc_open_to_buy 0.063405
0.121734
bc_util -0.130236
-0.001699
mo_sin_old_il_acct -0.004021
-0.096722
mo_sin_old_rev_tl_op -0.024198
-0.106522
mo_sin_rcnt_rev_tl_op -0.433578
-0.037148
mo_sin_rcnt_tl -0.534940
-0.030447
mort_acc 0.067999
-0.045008
mths_since_recent_bc -0.287923
-0.054641
mths_since_recent_inq -0.225449
0.017137
num_accts_ever_120_pd 0.057674
-0.550597
num_actv_bc_tl 0.083757
0.127743
num_actv_rev_tl 0.177864
0.096877
num_bc_sats 0.140359
0.137486
num_bc_tl 0.155990
0.018570
num_il_tl 0.196592
-0.014351
num_op_rev_tl 0.257699
0.106137
num_rev_accts 0.226864
0.004681
num_rev_tl_bal_gt_0 0.178580
0.097005
num_tl_op_past_12m 1.000000
0.015170
pct_tl_nvr_dlq 0.015170
1.000000
percent_bc_gt_75 -0.123374
0.009183
pub_rec_bankruptcies -0.002445
0.033255

```
tot_hi_cred_lim                      0.100894
0.008941
total_bc_limit                       0.004563
0.191606
total_il_high_credit_limit           0.151449
-0.009095

                           percent_bc_gt_75  pub_re
c_bankruptcies  \
loan_amnt                          0.007203
-0.094852
funded_amnt                        0.007233
-0.094798
funded_amnt_inv                    0.007298
-0.094606
int_rate                           0.353748
0.048524
installment                        0.036409
-0.088497
annual_inc                        -0.038702
-0.054342
dti                                0.188117
-0.054327
delinq_2yrs                       -0.021144
-0.038079
inq_last_6mths                    -0.078196
0.004385
open_acc                          -0.092029
-0.048503
pub_rec                           -0.025297
0.759816
revol_bal                          0.087455
-0.105968
total_acc                         -0.079553
-0.020924
total_pymnt                        0.029349
-0.083643
total_pymnt_inv                    0.029401
-0.083494
total_rec_prncp                   -0.024151
-0.086063
total_rec_int                      0.135720
-0.050333
total_rec_late_fee                 0.026717
```

-0.015232
recoveries 0.040779
-0.013886
collection_recovery_fee 0.033718
-0.008255
last_pymnt_amnt -0.026677
-0.021988
tot_coll_amt -0.028502
0.018787
total_rev_hi_lim -0.137056
-0.093797
acc_open_past_24mths -0.116840
0.011326
avg_cur_bal 0.016887
-0.060993
bc_open_to_buy -0.477066
-0.087191
bc_util 0.831412
-0.012362
mo_sin_old_il_acct 0.029971
0.042991
mo_sin_old_rev_tl_op -0.026080
0.035970
mo_sin_rcnt_rev_tl_op 0.086981
-0.033634
mo_sin_rcnt_tl 0.081118
-0.012071
mort_acc -0.039093
0.000999
mths_since_recent_bc 0.123060
-0.009216
mths_since_recent_inq 0.055103
-0.006782
num_accts_ever_120_pd -0.027612
-0.004829
num_actv_bc_tl 0.040894
-0.037957
num_actv_rev_tl 0.098692
-0.023378
num_bc_sats -0.177885
-0.046185
num_bc_tl -0.154246
-0.015137
num_il_tl 0.027746
-0.023903

```
num_op_rev_tl                        -0.128804
-0.026013
num_rev_accts                        -0.132262
-0.014779
num_rev_tl_bal_gt_0                   0.098695
-0.023540
num_tl_op_past_12m                   -0.123374
-0.002445
pct_tl_nvr_dlq                        0.009183
0.033255
percent_bc_gt_75                      1.000000
-0.016911
pub_rec_bankruptcies                 -0.016911
1.000000
tot_hi_cred_lim                      -0.048225
-0.088544
total_bc_limit                       -0.249861
-0.142249
total_il_high_credit_limit            0.010729
-0.039593

                          tot_hi_cred_lim  total_b
c_limit  \
loan_amnt                        0.306396        0
.358044
funded_amnt                      0.306465        0
.358130
funded_amnt_inv                  0.306440        0
.358048
int_rate                        -0.155279       -0
.261619
installment                      0.270641        0
.323503
annual_inc                       0.481292        0
.288585
dti                             -0.004814        0
.031969
delinq_2yrs                      0.080387       -0
.059412
inq_last_6mths                   0.097522        0
.009950
open_acc                         0.255636        0
.301572
pub_rec                         -0.063181       -0
```

.116149
revol_bal 0.448612 0
.478627
total_acc 0.327509 0
.257766
total_pymnt 0.274987 0
.309217
total_pymnt_inv 0.274958 0
.309131
total_rec_prncp 0.296321 0
.348431
total_rec_int 0.140758 0
.125307
total_rec_late_fee 0.018769 -0
.008083
recoveries 0.019262 0
.011394
collection_recovery_fee 0.023102 0
.012161
last_pymnt_amnt 0.169447 0
.155508
tot_coll_amt -0.002933 -0
.031221
total_rev_hi_lim 0.447925 0
.561239
acc_open_past_24mths 0.101169 -0
.009808
avg_cur_bal 0.822000 0
.151672
bc_open_to_buy 0.233889 0
.839838
bc_util -0.045084 -0
.290919
mo_sin_old_il_acct 0.181827 0
.104874
mo_sin_old_rev_tl_op 0.205132 0
.265807
mo_sin_rcnt_rev_tl_op 0.027206 0
.007140
mo_sin_rcnt_tl -0.075786 0
.010811
mort_acc 0.512527 0
.220792
mths_since_recent_bc 0.040567 -0
.062443

```
mths_since_recent_inq              -0.059908        0
.003645
num_accts_ever_120_pd               0.006180       -0
.123785
num_actv_bc_tl                      0.060523        0
.369297
num_actv_rev_tl                     0.059402        0
.195929
num_bc_sats                         0.103953        0
.520473
num_bc_tl                           0.153889        0
.396479
num_il_tl                           0.201089       -0
.016023
num_op_rev_tl                       0.111189        0
.340577
num_rev_accts                       0.180928        0
.313148
num_rev_tl_bal_gt_0                 0.059625        0
.196284
num_tl_op_past_12m                  0.100894        0
.004563
pct_tl_nvr_dlq                      0.008941        0
.191606
percent_bc_gt_75                   -0.048225       -0
.249861
pub_rec_bankruptcies               -0.088544       -0
.142249
tot_hi_cred_lim                     1.000000        0
.353012
total_bc_limit                      0.353012        1
.000000
total_il_high_credit_limit          0.375659        0
.072251

                           total_il_high_credit_lim
it
loan_amnt                                  0.1738
11
funded_amnt                                0.1738
21
funded_amnt_inv                            0.1738
94
int_rate                                   0.0248
```

08
installment 0.1649
82
annual_inc 0.2923
26
dti 0.3225
81
delinq_2yrs 0.0685
56
inq_last_6mths 0.0948
96
open_acc 0.3341
69
pub_rec -0.0238
02
revol_bal 0.0944
52
total_acc 0.3821
61
total_pymnt 0.1585
47
total_pymnt_inv 0.1586
09
total_rec_prncp 0.1516
23
total_rec_int 0.1204
54
total_rec_late_fee 0.0250
26
recoveries 0.0338
33
collection_recovery_fee 0.0329
16
last_pymnt_amnt 0.0978
72
tot_coll_amt 0.0038
19
total_rev_hi_lim 0.0687
52
acc_open_past_24mths 0.2006
55
avg_cur_bal 0.1883
51
bc_open_to_buy 0.0272
64

bc_util 0.0194
64
mo_sin_old_il_acct 0.1794
61
mo_sin_old_rev_tl_op 0.0187
94
mo_sin_rcnt_rev_tl_op 0.0051
09
mo_sin_rcnt_tl -0.1201
61
mort_acc 0.1050
41
mths_since_recent_bc 0.0072
51
mths_since_recent_inq -0.0526
63
num_accts_ever_120_pd 0.0417
45
num_actv_bc_tl -0.0101
72
num_actv_rev_tl -0.0122
39
num_bc_sats 0.0028
15
num_bc_tl 0.0331
86
num_il_tl 0.6056
54
num_op_rev_tl 0.0021
44
num_rev_accts 0.0430
24
num_rev_tl_bal_gt_0 -0.0121
11
num_tl_op_past_12m 0.1514
49
pct_tl_nvr_dlq -0.0090
95
percent_bc_gt_75 0.0107
29
pub_rec_bankruptcies -0.0395
93
tot_hi_cred_lim 0.3756
59

```
total_bc_limit                                0.0722
51
total_il_high_credit_limit                    1.0000
00

[50 rows x 50 columns]
```

In [66]:

```
numDF = numDF[selected_columns]
```

In [67]:

```
# £print(nonnumDF[['grade','home_ownership','loan_status','purpo
se','sub_grade','term']])
```

In [68]:

```
# Quartiles and IQR for mths_since_recent_bc

for col in numDF.columns:
    quartiles = numDF[col].quantile([0.25, 0.75], interpolation=
'nearest')
    q1 = quartiles[0.25]
    q3 = quartiles[0.75]
    IQR = q3 - q1
    outlier_val = q3 + 1.5*IQR
    #print("Outlier val:", outlier_val)
    numDF[col] = np.where(numDF[col] > outlier_val, outlier_val,
numDF[col])

#df1 = df['mths_since_recent_bc'] = np.where(df['mths_since_rece
nt_bc'] > outlier_val, outlier_val, df['mths_since_recent_bc'])
```

In [69]:

```
nonnumDF.columns
```

Out[69]:

```
Index(['term', 'grade', 'sub_grade', 'home_ownership
', 'verification_status',
       'issue_d', 'loan_status', 'pymnt_plan', 'desc
', 'purpose',
       'earliest_cr_line', 'revol_util', 'initial_li
st_status', 'last_pymnt_d',
       'last_credit_pull_d'],
      dtype='object')
```

In [70]:

```
finaldf = pd.concat([numDF, nonnumDF[['grade','home_ownership','
loan_status','purpose','term','sub_grade','verification_status']
]], axis=1)
```

In [71]:

```
finaldf_mn = pd.concat([numDF, nonnumDF[['grade','home_ownership
','loan_status','purpose','term','sub_grade','verification_statu
s']]], axis=1)
```

In [72]:

```
np.unique(finaldf['loan_status'])
```

Out[72]:

```
array(['Charged Off', 'Fully Paid'], dtype=object)
```

In [73]:

```
finaldf["loan_status"] = np.where(finaldf["loan_status"] == "Cha
rged Off", 1, 0)
```

In [74]:

```
finaldf_mn["loan_status"] = np.where(finaldf_mn["loan_status"] =
= "Charged Off", 1, 0)
```

In [75]:

```
print('Number of missing values per column:')
countMissing = finaldf_mn['loan_status']
count = 0
ncount = 0
for w in countMissing:
    if w == 1:
        count+=1
    else:
        ncount+=1
print(count/len(finaldf_mn['loan_status']))
print(ncount/len(finaldf_mn['loan_status']))
```

```
Number of missing values per column:
0.15767266620965986
0.8423273337903402
```

In [ ]:

```

```

In [76]:

```
## Undersampling because of imbalanced dataset
Chargedoff = len(finaldf_mn[finaldf_mn['loan_status'] == 1])
Paid_indices = finaldf_mn[finaldf_mn.loan_status == 0].index
random_indices = np.random.choice(Paid_indices,Chargedoff, repla
ce=False)
Chargedoff_indices = finaldf_mn[finaldf_mn.loan_status == 1].ind
ex
under_sample_indices = np.concatenate([Chargedoff_indices,random
_indices])
finaldf_mn = finaldf_mn.loc[under_sample_indices]
# finaldf_main['loan_status']
#print(Chargedoff)
```

In [77]:

```
## Undersampling because of imbalanced dataset
Chargedoff = len(finaldf[finaldf['loan_status'] == 1])
Paid_indices = finaldf[finaldf.loan_status == 0].index
random_indices = np.random.choice(Paid_indices,Chargedoff, repla
ce=False)
Chargedoff_indices = finaldf[finaldf.loan_status == 1].index
under_sample_indices = np.concatenate([Chargedoff_indices,random
_indices])
finaldf = finaldf.loc[under_sample_indices]
# finaldf['loan_status']
#print(Chargedoff)
```

In [78]:

```
# Selecting categorical variables based on chi2 test
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
finaldf['grade']  = finaldf['grade'].astype('str')
finaldf['home_ownership']  = finaldf['home_ownership'].astype('s
tr')
finaldf['purpose']  = finaldf['purpose'].astype('str')
finaldf['term']  = finaldf['term'].astype('str')
finaldf['sub_grade']  = finaldf['sub_grade'].astype('str')
finaldf['verification_status']  = finaldf['verification_status']
.astype('str')


finaldf['grade'] = label_encoder.fit_transform(finaldf['grade'])
finaldf['home_ownership'] = label_encoder.fit_transform(finaldf[
'home_ownership'])
finaldf['purpose'] = label_encoder.fit_transform(finaldf['purpos
e'])
finaldf['term'] = label_encoder.fit_transform(finaldf['term'])
finaldf['sub_grade'] = label_encoder.fit_transform(finaldf['sub_
grade'])
finaldf['verification_status'] = label_encoder.fit_transform(fin
aldf['verification_status'])
```

In [79]:

```
from sklearn.feature_selection import chi2
chi_scores = chi2(finaldf[['grade','home_ownership','purpose','t
erm','sub_grade','verification_status']],finaldf['loan_status'])
```

In [80]:

```
chi_scores
```

Out[80]:

```
(array([ 3916.51881998,   321.17133001,   248.920361
59,  1600.00571755,
        17191.24455441,   323.46331733]),
 array([0.00000000e+00, 8.04936533e-72, 4.46490874e-
56, 0.00000000e+00,
        0.00000000e+00, 2.54992977e-72]))
```

In [81]:

```
import numpy as np
from sklearn.feature_selection import SelectKBest, f_classif,chi
2
from sklearn.preprocessing import LabelEncoder

predictors = ['grade','home_ownership','purpose','term','sub_gra
de','verification_status']


# Perform feature selection
selector = SelectKBest(chi2, k='all')
selector.fit(finaldf, finaldf['loan_status'])

# Get the raw p-values for each feature, and transform from p-va
lues into scores
scores = selector.scores_
print(scores)
# Plot the scores.  See how "Pclass", "Sex", "Title", and "Fare"
are the best?
# plt.bar(range(len(predictors)), scores)
# plt.xticks(range(len(predictors)), predictors, rotation='verti
cal')
# plt.show()
```

```
[1.64254588e+06 6.38815215e+03 8.26039419e+06 2.3509
1351e+03
            nan 4.29675327e+02 4.89342920e+01
nan
 8.48236435e+04 9.27092466e+01            nan
nan
 9.23950242e+07            nan 1.99136345e+06 1.0266
9607e+03
 3.45998425e+06 4.93849023e+06 4.26101196e+03 7.8656
9796e+02
 5.04817503e+03 1.97804989e+03 1.37252174e+03 4.3635
4981e+02
 5.62182132e+03 1.12787413e+03 4.54951947e+00 5.4885
7982e+01
 3.03663345e+02 9.02943303e+00 1.10957385e+01 9.5337
7200e+01
 3.67138267e+02 2.34375085e+00            nan 1.2209
8862e+05
 3.91651882e+03 3.21171330e+02 2.96710000e+04 2.4892
0362e+02
 1.60000572e+03 1.71912446e+04 3.23463317e+02]
```

In [ ]:

```
```

In [33]:

```
## Dividing dataset into 2 parts
x_train, x_test, y_train, y_test = train_test_split(finaldf_mn.l
oc[:, finaldf_mn.columns != 'loan_status'], finaldf_mn['loan_sta
tus'], test_size=0.3, random_state = 0)
```

In [ ]:

```
```

In [82]:

```
x_train_sc = x_train[['loan_amnt', 'int_rate', 'annual_inc', 'dt
i','revol_bal','last_pymnt_amnt','total_rev_hi_lim','avg_cur_bal
',
       'bc_open_to_buy', 'bc_util']]
x_test_sc = x_test[['loan_amnt', 'int_rate', 'annual_inc', 'dti'
,'revol_bal','last_pymnt_amnt','total_rev_hi_lim','avg_cur_bal',
       'bc_open_to_buy', 'bc_util']]
```

In [83]:

```
scaler = StandardScaler()
scaler.fit(x_train_sc)
scaler.fit(x_test_sc)
names = x_train_sc.columns
x_train_scaled = scaler.transform(x_train_sc)
x_train_scaled = pd.DataFrame(x_train_scaled, columns=names)
x_test_scaled = scaler.transform(x_test_sc)
x_test_scaled = pd.DataFrame(x_test_scaled, columns=names)
```

In [84]:

```
x_train
```

Out[84]:

| | loan_amnt | int_rate | annual_inc | dti | delinq_2yrs | inq_last_6mth |
|---|---|---|---|---|---|---|
| **0** | 11075.0 | 13.11 | 32000.0 | 17.93 | 0.0 | 0. |
| **1** | 21000.0 | 12.12 | 100000.0 | 17.74 | 0.0 | 0. |
| **2** | 2650.0 | 16.78 | 22800.0 | 12.00 | 0.0 | 1. |
| **3** | 4000.0 | 13.68 | 41000.0 | 22.62 | 0.0 | 2. |
| **4** | 8025.0 | 23.28 | 37235.0 | 23.11 | 0.0 | 2. |
| **5** | 2350.0 | 15.81 | 30000.0 | 29.48 | 0.0 | 2. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **6** | 5000.0 | 18.75 | 54000.0 | 15.98 | 0.0 | 2. |
| **7** | 12800.0 | 13.99 | 65000.0 | 14.86 | 0.0 | 2. |
| **8** | 2000.0 | 17.27 | 25000.0 | 24.41 | 0.0 | 0. |
| **9** | 5375.0 | 9.71 | 41000.0 | 19.40 | 0.0 | 0. |
| **10** | 35000.0 | 24.99 | 85000.0 | 21.16 | 0.0 | 1. |
| **11** | 30000.0 | 22.95 | 76500.0 | 30.71 | 0.0 | 1. |
| **12** | 15000.0 | 12.12 | 38000.0 | 3.41 | 0.0 | 0. |
| **13** | 2000.0 | 12.12 | 30000.0 | 16.08 | 0.0 | 0. |
| **14** | 12250.0 | 16.29 | 43000.0 | 16.05 | 0.0 | 0. |
| **15** | 13475.0 | 13.11 | 47000.0 | 23.92 | 0.0 | 0. |
| **16** | 30000.0 | 10.65 | 105996.0 | 20.22 | 0.0 | 1. |
| **17** | 14250.0 | 11.14 | 53400.0 | 13.69 | 0.0 | 1. |
| **18** | 20000.0 | 15.61 | 50000.0 | 11.51 | 0.0 | 0. |
| **19** | 13000.0 | 14.33 | 61000.0 | 19.36 | 0.0 | 2. |
| **20** | 35000.0 | 16.29 | 82000.0 | 30.01 | 0.0 | 0. |
| **21** | 10000.0 | 22.20 | 140000.0 | 4.47 | 0.0 | 2. |
| **22** | 15875.0 | 13.11 | 55000.0 | 24.96 | 0.0 | 2. |
| **23** | 8700.0 | 11.99 | 65000.0 | 5.72 | 0.0 | 0. |
| **24** | 10625.0 | 12.12 | 40000.0 | 26.52 | 0.0 | 0. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **25** | 12000.0 | 11.14 | 106000.0 | 19.20 | 0.0 | 0. |
| **26** | 12000.0 | 17.56 | 42000.0 | 28.57 | 0.0 | 2. |
| **27** | 3750.0 | 20.80 | 21432.0 | 24.30 | 0.0 | 0. |
| **28** | 8000.0 | 20.49 | 58000.0 | 7.24 | 0.0 | 2. |
| **29** | 20000.0 | 22.78 | 150000.0 | 1.63 | 0.0 | 0. |
| **...** | ... | ... | ... | ... | ... | . |
| **41509** | 10000.0 | 6.62 | 45000.0 | 9.41 | 0.0 | 2. |
| **41510** | 20000.0 | 18.55 | 85000.0 | 17.04 | 0.0 | 1. |
| **41511** | 6175.0 | 17.77 | 29952.0 | 8.21 | 0.0 | 0. |
| **41512** | 7200.0 | 13.67 | 52000.0 | 32.10 | 0.0 | 2. |
| **41513** | 8000.0 | 18.49 | 80000.0 | 18.60 | 0.0 | 1. |
| **41514** | 8000.0 | 16.29 | 50000.0 | 17.04 | 0.0 | 1. |
| **41515** | 32000.0 | 14.30 | 150000.0 | 7.32 | 0.0 | 0. |
| **41516** | 10000.0 | 19.22 | 48000.0 | 24.35 | 0.0 | 0. |
| **41517** | 12075.0 | 13.11 | 65000.0 | 32.76 | 0.0 | 2. |
| **41518** | 8000.0 | 11.99 | 50000.0 | 21.17 | 0.0 | 0. |
| **41519** | 14400.0 | 12.12 | 67000.0 | 15.39 | 0.0 | 0. |
| **41520** | 26000.0 | 18.75 | 150000.0 | 6.66 | 0.0 | 0. |
| **41521** | 17500.0 | 10.74 | 55000.0 | 9.97 | 0.0 | 0. |

| | | | | | | |
|---|---|---|---|---|---|---|
| **41522** | 8000.0 | 21.00 | 28000.0 | 27.90 | 0.0 | 2. |
| **41523** | 20000.0 | 8.90 | 58000.0 | 31.76 | 0.0 | 0. |
| **41524** | 11000.0 | 12.12 | 45000.0 | 9.07 | 0.0 | 0. |
| **41525** | 19600.0 | 21.98 | 150000.0 | 5.74 | 0.0 | 0. |
| **41526** | 5600.0 | 13.67 | 43000.0 | 14.06 | 0.0 | 0. |
| **41527** | 10000.0 | 10.99 | 95000.0 | 14.03 | 0.0 | 1. |
| **41528** | 20000.0 | 12.12 | 49000.0 | 28.53 | 0.0 | 0. |
| **41529** | 8250.0 | 12.35 | 60000.0 | 10.54 | 0.0 | 1. |
| **41530** | 18000.0 | 16.29 | 48000.0 | 18.26 | 0.0 | 1. |
| **41531** | 5000.0 | 15.61 | 68000.0 | 12.87 | 0.0 | 1. |
| **41532** | 15000.0 | 13.99 | 52500.0 | 4.96 | 0.0 | 0. |
| **41533** | 16000.0 | 14.33 | 65640.0 | 12.10 | 0.0 | 0. |
| **41534** | 15000.0 | 18.85 | 100000.0 | 23.04 | 0.0 | 1. |
| **41535** | 20000.0 | 12.12 | 50000.0 | 19.72 | 0.0 | 2. |
| **41536** | 15000.0 | 17.76 | 45000.0 | 20.48 | 0.0 | 1. |
| **41537** | 9950.0 | 17.56 | 60000.0 | 14.10 | 0.0 | 1. |
| **41538** | 16000.0 | 19.20 | 77515.0 | 17.26 | 0.0 | 0. |

In [85]:

```
x_train=x_train.reset_index()
#x_train = x_train.drop('index')
x_train = x_train.loc[:, x_train.columns != 'index']
```

In [86]:

```
x_test=x_test.reset_index()
#x_train = x_train.drop('index')
x_test = x_test.loc[:, x_test.columns != 'index']
```

In [87]:

```
x_train_scaled = pd.concat([x_train_scaled[['loan_amnt', 'int_rate', 'annual_inc', 'dti','revol_bal','last_pymnt_amnt','total_rev_hi_lim','avg_cur_bal',
       'bc_open_to_buy', 'bc_util']], x_train[['sub_grade','home_ownership','purpose','term']]], axis=1)
```

In [88]:

```
x_test_scaled = pd.concat([x_test_scaled[['loan_amnt', 'int_rate', 'annual_inc', 'dti','revol_bal','last_pymnt_amnt','total_rev_hi_lim','avg_cur_bal',
       'bc_open_to_buy', 'bc_util']], x_test[['sub_grade','home_ownership','purpose','term']]], axis=1)
```

In [89]:

```
#making categorical variable dummy variable
x_train_scaled = pd.get_dummies(x_train_scaled, columns=['home_ownership','purpose','term','sub_grade'])
```

In [90]:

```
#making categorical variable dummy variable
x_test_scaled = pd.get_dummies(x_test_scaled, columns=['home_ownership','purpose','term','sub_grade'])
```

In [91]:

```
from sklearn import model_selection
from sklearn.ensemble import RandomForestClassifier
# random forest model creation
rfc = RandomForestClassifier()
rfc.fit(x_train_scaled,y_train)
# predictions
rfc_predict = rfc.predict(x_test_scaled)
```

```
/Users/michellebaginski/anaconda3/lib/python3.7/site
-packages/sklearn/ensemble/forest.py:245: FutureWarn
ing: The default value of n_estimators will change f
rom 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarnin
g)
```

In [92]:

```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report, confusion_mat
rix
```

In [93]:

```
rfc_cv_score = cross_val_score(rfc, x_train_scaled, y_train, cv=
10, scoring='roc_auc')
```

In [94]:

```
print("=== Confusion Matrix ===")
print(confusion_matrix(y_test, rfc_predict))
print('\n')
print("=== Classification Report ===")
print(classification_report(y_test, rfc_predict))
print('\n')
print("=== All AUC Scores ===")
print(rfc_cv_score)
print('\n')
print("=== Mean AUC Score ===")
print("Mean AUC Score - Random Forest: ", rfc_cv_score.mean())
```

```
=== Confusion Matrix ===
[[6808 2148]
 [1859 6988]]


=== Classification Report ===
              precision    recall  f1-score   suppor
t

           0       0.79      0.76      0.77       895
6
           1       0.76      0.79      0.78       884
7

    accuracy                           0.77      1780
3
   macro avg       0.78      0.78      0.77      1780
3
weighted avg       0.78      0.77      0.77      1780
3



=== All AUC Scores ===
[0.86952638 0.86273777 0.8465039  0.86014253 0.84162
513 0.85273836
 0.85622041 0.86176969 0.86549607 0.86477387]


=== Mean AUC Score ===
Mean AUC Score - Random Forest:  0.8581534094395709
```

In [95]:

```
rfc = RandomForestClassifier(n_estimators=600, max_depth=300, ma
x_features='sqrt')
rfc.fit(x_train_scaled,y_train)
rfc_predict = rfc.predict(x_test_scaled)
rfc_cv_score = cross_val_score(rfc, x_train_scaled, y_train, cv=
10, scoring='roc_auc')
print("=== Confusion Matrix ===")
print(confusion_matrix(y_test, rfc_predict))
print('\n')
print("=== Classification Report ===")
print(classification_report(y_test, rfc_predict))
print('\n')
print("=== All AUC Scores ===")
print(rfc_cv_score)
print('\n')
print("=== Mean AUC Score ===")
print("Mean AUC Score - Random Forest: ", rfc_cv_score.mean())
```

```
=== Confusion Matrix ===
[[6439 2517]
 [ 941 7906]]


=== Classification Report ===
              precision    recall  f1-score   suppor
t

           0       0.87      0.72      0.79       895
6
           1       0.76      0.89      0.82       884
7

    accuracy                           0.81      1780
3
   macro avg       0.82      0.81      0.80      1780
3
weighted avg       0.82      0.81      0.80      1780
3



=== All AUC Scores ===
[0.88882677 0.88471159 0.87540674 0.8839003  0.86899
906 0.87545567
 0.88477481 0.8874521  0.89123241 0.88543022]


=== Mean AUC Score ===
Mean AUC Score - Random Forest:  0.882618967205401
```

In [96]:

```
for name, importance in zip(x_train_scaled, rfc.feature_importan
ces_):
    print(name, "=", importance)
```

```
loan_amnt = 0.053761735402326115
int_rate = 0.06534370526283916
annual_inc = 0.054183461727034365
dti = 0.058000113452690065
revol_bal = 0.05403827061955187
last_pymnt_amnt = 0.3890238634759968
```

total_rev_hi_lim = 0.04660280187282706
avg_cur_bal = 0.04980539790538859
bc_open_to_buy = 0.05199553285759633
bc_util = 0.0525780966143542
home_ownership_MORTGAGE = 0.006340654698363726
home_ownership_NONE = 0.00010530702127745545
home_ownership_OTHER = 9.771610627428151e-05
home_ownership_OWN = 0.0043011820447871885
home_ownership_RENT = 0.006627023612486046
purpose_car = 0.0012395108156892766
purpose_credit_card = 0.006003184493149518
purpose_debt_consolidation = 0.007169357383117477
purpose_home_improvement = 0.0032070390840867975
purpose_house = 0.0007942808990084938
purpose_major_purchase = 0.0018589426289013 74
purpose_medical = 0.0011033730401453033
purpose_moving = 0.0008226783482957738
purpose_other = 0.003211154227492836
purpose_renewable_energy = 0.00010885846394341059
purpose_small_business = 0.00201868959261254
purpose_vacation = 0.0007236544636986772
purpose_wedding = 0.0008521695712419 87
term_ 36 months = 0.0125991524955235 64
term_ 60 months = 0.014073781766196 1
sub_grade_A1 = 0.0027883300854814057
sub_grade_A2 = 0.0018593853916773 49
sub_grade_A3 = 0.0019715629018917627
sub_grade_A4 = 0.0022055367409211 36
sub_grade_A5 = 0.0022532047551360 94
sub_grade_B1 = 0.0024917090924836066
sub_grade_B2 = 0.0025589170266725235
sub_grade_B3 = 0.0027200110043401147
sub_grade_B4 = 0.0025851132079777288
sub_grade_B5 = 0.0022954789974467 25
sub_grade_C1 = 0.0025511333212010715
sub_grade_C2 = 0.0025088912976853174
sub_grade_C3 = 0.0023110296081088485
sub_grade_C4 = 0.0022828778471571 86
sub_grade_C5 = 0.0021872793012298 47
sub_grade_D1 = 0.0020516435105572247
sub_grade_D2 = 0.0017419105956409 09
sub_grade_D3 = 0.0016294641891568288
sub_grade_D4 = 0.0016815565074051097
sub_grade_D5 = 0.0015432221089399696
sub_grade_E1 = 0.0010411085434722067

sub_grade_E2 = 0.0011265212760308938
sub_grade_E3 = 0.0008356202692167288
sub_grade_E4 = 0.0008561076254123023
sub_grade_E5 = 0.0007876314085303913
sub_grade_F1 = 0.0006435984466996988
sub_grade_F2 = 0.0004964376397609544
sub_grade_F3 = 0.0004861403605330628
sub_grade_F4 = 0.0004265515762155708
sub_grade_F5 = 0.00029081062283887164
sub_grade_G1 = 0.00021227894734017263
sub_grade_G2 = 0.00013392389980012922
sub_grade_G3 = 0.0001008456548722498
sub_grade_G4 = 6.978771921643279e-05
sub_grade_G5 = 4.047605967877071e-05