Problem Statement

Solving this assignment will give you an idea about how real business problems are solved usingEDA. In this case study, apart from applying the techniques you have learnt in EDA, you will alsodevelop a basic understanding of risk analytics in banking and financial services and understand how data is used to minimise the risk of losing money while lending to customers.

>Import Libraries

In [1]:

```
#import usefull libraries like Pandas and Numpy to read csv-files
import pandas as pd
import numpy as np
```

>Read Dataset

In [2]:

```
#Load Loan data file
loan_file = pd.read_csv('loan.csv')
loan_file
```

```
D:\Anaconda\lib\site-packages\IPython\core\interactiveshell.py:2785: DtypeWarning: Columns (47) have mixed types. Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[2]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	..
0	1077501	1296599	5000	5000	4975.00000	36 months	10.65%	162.87	B	B2	..
1	1077430	1314167	2500	2500	2500.00000	60 months	15.27%	59.83	C	C4	..
2	1077175	1313524	2400	2400	2400.00000	36 months	15.96%	84.33	C	C5	..
3	1076863	1277178	10000	10000	10000.00000	36 months	13.49%	339.31	C	C1	..
4	1075358	1311748	3000	3000	3000.00000	60 months	12.69%	67.79	B	B5	..
5	1075269	1311441	5000	5000	5000.00000	36 months	7.90%	156.46	A	A4	..
6	1069639	1304742	7000	7000	7000.00000	60 months	15.96%	170.08	C	C5	..
7	1072053	1288686	3000	3000	3000.00000	36 months	18.64%	109.43	E	E1	..
8	1071795	1306957	5600	5600	5600.00000	60 months	21.28%	152.39	F	F2	..
9	1071570	1306721	5375	5375	5350.00000	60 months	12.69%	121.45	B	B5	..
10	1070078	1305201	6500	6500	6500.00000	60 months	14.65%	153.45	C	C3	..
11	1069908	1305008	12000	12000	12000.00000	36 months	12.69%	402.54	B	B5	..
						36					

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...
12	1064687	1298717	9000	9000	9000.00000	36 months	13.49%	305.38	C	C1	...
13	1069866	1304956	3000	3000	3000.00000	36 months	9.91%	96.68	B	B1	...
14	1069057	1303503	10000	10000	10000.00000	36 months	10.65%	325.74	B	B2	...
15	1069759	1304871	1000	1000	1000.00000	36 months	16.29%	35.31	D	D1	...
16	1065775	1299699	10000	10000	10000.00000	36 months	15.27%	347.98	C	C4	...
17	1069971	1304884	3600	3600	3600.00000	36 months	6.03%	109.57	A	A1	...
18	1062474	1294539	6000	6000	6000.00000	36 months	11.71%	198.46	B	B3	...
19	1069742	1304855	9200	9200	9200.00000	36 months	6.03%	280.01	A	A1	...
20	1069740	1284848	20250	20250	19142.16108	60 months	15.27%	484.63	C	C4	...
21	1039153	1269083	21000	21000	21000.00000	36 months	12.42%	701.73	B	B4	...
22	1069710	1304821	10000	10000	10000.00000	36 months	11.71%	330.76	B	B3	...
23	1069700	1304810	10000	10000	10000.00000	36 months	11.71%	330.76	B	B3	...
24	1069559	1304634	6000	6000	6000.00000	36 months	11.71%	198.46	B	B3	...
25	1069697	1273773	15000	15000	15000.00000	36 months	9.91%	483.38	B	B1	...
26	1069800	1304679	15000	15000	8725.00000	36 months	14.27%	514.64	C	C2	...
27	1069657	1304764	5000	5000	5000.00000	60 months	16.77%	123.65	D	D2	...
28	1069799	1304678	4000	4000	4000.00000	36 months	11.71%	132.31	B	B3	...
29	1047704	1278806	8500	8500	8500.00000	36 months	11.71%	281.15	B	B3	...
...	...	...	...	...	...	...	...	...	...	...	...
39687	111307	105982	12000	12000	2500.00000	36 months	12.49%	401.37	D	D3	...
39688	111227	111223	20000	20000	2800.00000	36 months	13.43%	678.08	E	E1	...
39689	109355	109346	1200	1200	0.00000	36 months	11.54%	39.60	C	C5	...
39690	107136	107130	12250	12250	1525.00000	36 months	10.59%	398.69	C	C2	...
39691	106360	106333	2700	2700	550.00000	36 months	15.96%	94.88	F	F4	...
39692	76597	76583	5000	5000	1775.00000	36 months	9.01%	159.03	B	B2	...
39693	106079	106039	3500	3500	1200.00000	36 months	9.96%	112.87	B	B5	...
39694	90966	90962	5000	5000	4150.00000	36 months	7.43%	155.38	A	A2	...
39695	93449	93423	5000	5000	3100.00000	36	7.43%	155.38	A	A2	

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...
39695	92440	92423	5000	5000	3100.00000	months	7.43%	155.38	A	A2	...
39696	102376	95212	25000	25000	525.00000	36 months	10.59%	813.65	C	C2	...
39697	101579	100083	10000	10000	400.00000	36 months	10.28%	323.98	C	C1	...
39698	98982	98957	5000	5000	675.00000	36 months	9.01%	159.03	B	B2	...
39699	98339	97572	5100	5100	575.00000	36 months	8.38%	160.72	A	A5	...
39700	98276	98268	5400	5400	200.00000	36 months	7.75%	168.60	A	A3	...
39701	96844	95222	5300	5300	600.00000	36 months	8.38%	167.02	A	A5	...
39702	96350	96338	5000	5000	850.00000	36 months	11.22%	164.23	C	C4	...
39703	94838	73673	3000	3000	2550.00000	36 months	10.28%	97.20	C	C1	...
39704	93277	93254	3000	3000	950.00000	36 months	8.70%	94.98	B	B1	...
39705	93061	93057	5000	5000	250.00000	36 months	7.43%	155.38	A	A2	...
39706	92676	92671	5000	5000	150.00000	36 months	8.07%	156.84	A	A4	...
39707	92666	92661	5000	5000	525.00000	36 months	9.33%	159.77	B	B3	...
39708	92552	92542	5000	5000	375.00000	36 months	9.96%	161.25	B	B5	...
39709	92533	92529	5000	5000	675.00000	36 months	11.22%	164.23	C	C4	...
39710	92507	92502	5000	5000	250.00000	36 months	7.43%	155.38	A	A2	...
39711	92402	92390	5000	5000	700.00000	36 months	8.70%	158.30	B	B1	...
39712	92187	92174	2500	2500	1075.00000	36 months	8.07%	78.42	A	A4	...
39713	90665	90607	8500	8500	875.00000	36 months	10.28%	275.38	C	C1	...
39714	90395	90390	5000	5000	1325.00000	36 months	8.07%	156.84	A	A4	...
39715	90376	89243	5000	5000	650.00000	36 months	7.43%	155.38	A	A2	...
39716	87023	86999	7500	7500	800.00000	36 months	13.75%	255.43	E	E2	...

39717 rows × 111 columns

Data Cleaning

> Delete unwanted columns

In [3]:

```
#Drop all column with null value
```

```
loan_new = loan_file.dropna(axis=1,how='all')
loan_new.shape
```

Out[3]:

```
(39717, 57)
```

In [4]:

```
#Remove column with only one unique values
loan_new=loan_new.loc[:,loan_new.nunique()!=1]
loan_new.shape
```

Out[4]:

```
(39717, 48)
```

In [5]:

```
# Drop columns with more than 50% null values

loan_new=loan_new.loc[:,round(loan_new.isnull().sum()/len(loan_new)*100,2)<50]
loan_new.shape
```

Out[5]:

```
(39717, 45)
```

In [6]:

```
# As we only want to find out potential defaults, we should remove 'current' from loan status

loan_new=loan_new[loan_new.loan_status !='Current']
loan_new=loan_new.loc[:,loan_new.nunique()!=1]
loan_file
```

Out[6]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	..
0	1077501	1296599	5000	5000	4975.00000	36 months	10.65%	162.87	B	B2	..
1	1077430	1314167	2500	2500	2500.00000	60 months	15.27%	59.83	C	C4	..
2	1077175	1313524	2400	2400	2400.00000	36 months	15.96%	84.33	C	C5	..
3	1076863	1277178	10000	10000	10000.00000	36 months	13.49%	339.31	C	C1	..
4	1075358	1311748	3000	3000	3000.00000	60 months	12.69%	67.79	B	B5	..
5	1075269	1311441	5000	5000	5000.00000	36 months	7.90%	156.46	A	A4	..
6	1069639	1304742	7000	7000	7000.00000	60 months	15.96%	170.08	C	C5	..
7	1072053	1288686	3000	3000	3000.00000	36 months	18.64%	109.43	E	E1	..
8	1071795	1306957	5600	5600	5600.00000	60 months	21.28%	152.39	F	F2	..
9	1071570	1306721	5375	5375	5350.00000	60 months	12.69%	121.45	B	B5	..
10	1070078	1305201	6500	6500	6500.00000	60 months	14.65%	153.45	C	C3	..

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...
11	1069908	1305008	12000	12000	12000.00000	36 months	12.69%	402.54	B	B5	...
12	1064687	1298717	9000	9000	9000.00000	36 months	13.49%	305.38	C	C1	...
13	1069866	1304956	3000	3000	3000.00000	36 months	9.91%	96.68	B	B1	...
14	1069057	1303503	10000	10000	10000.00000	36 months	10.65%	325.74	B	B2	...
15	1069759	1304871	1000	1000	1000.00000	36 months	16.29%	35.31	D	D1	...
16	1065775	1299699	10000	10000	10000.00000	36 months	15.27%	347.98	C	C4	...
17	1069971	1304884	3600	3600	3600.00000	36 months	6.03%	109.57	A	A1	...
18	1062474	1294539	6000	6000	6000.00000	36 months	11.71%	198.46	B	B3	...
19	1069742	1304855	9200	9200	9200.00000	36 months	6.03%	280.01	A	A1	...
20	1069740	1284848	20250	20250	19142.16108	60 months	15.27%	484.63	C	C4	...
21	1039153	1269083	21000	21000	21000.00000	36 months	12.42%	701.73	B	B4	...
22	1069710	1304821	10000	10000	10000.00000	36 months	11.71%	330.76	B	B3	...
23	1069700	1304810	10000	10000	10000.00000	36 months	11.71%	330.76	B	B3	...
24	1069559	1304634	6000	6000	6000.00000	36 months	11.71%	198.46	B	B3	...
25	1069697	1273773	15000	15000	15000.00000	36 months	9.91%	483.38	B	B1	...
26	1069800	1304679	15000	15000	8725.00000	36 months	14.27%	514.64	C	C2	...
27	1069657	1304764	5000	5000	5000.00000	60 months	16.77%	123.65	D	D2	...
28	1069799	1304678	4000	4000	4000.00000	36 months	11.71%	132.31	B	B3	...
29	1047704	1278806	8500	8500	8500.00000	36 months	11.71%	281.15	B	B3	...
...	...	...	...	...	...	...	...	...	...	...	...
39687	111307	105982	12000	12000	2500.00000	36 months	12.49%	401.37	D	D3	...
39688	111227	111223	20000	20000	2800.00000	36 months	13.43%	678.08	E	E1	...
39689	109355	109346	1200	1200	0.00000	36 months	11.54%	39.60	C	C5	...
39690	107136	107130	12250	12250	1525.00000	36 months	10.59%	398.69	C	C2	...
39691	106360	106333	2700	2700	550.00000	36 months	15.96%	94.88	F	F4	...
39692	76597	76583	5000	5000	1775.00000	36 months	9.01%	159.03	B	B2	...
39693	106079	106039	3500	3500	1200.00000	36 months	9.96%	112.87	B	B5	...
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	36	[illegible]	[illegible]	[illegible]	[illegible]	...

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...
39694	90966	90962	5000	5000	4150.00000	months	7.43%	155.38	A	A2	...
39695	92440	92423	5000	5000	3100.00000	36 months	7.43%	155.38	A	A2	...
39696	102376	95212	25000	25000	525.00000	36 months	10.59%	813.65	C	C2	...
39697	101579	100083	10000	10000	400.00000	36 months	10.28%	323.98	C	C1	...
39698	98982	98957	5000	5000	675.00000	36 months	9.01%	159.03	B	B2	...
39699	98339	97572	5100	5100	575.00000	36 months	8.38%	160.72	A	A5	...
39700	98276	98268	5400	5400	200.00000	36 months	7.75%	168.60	A	A3	...
39701	96844	95222	5300	5300	600.00000	36 months	8.38%	167.02	A	A5	...
39702	96350	96338	5000	5000	850.00000	36 months	11.22%	164.23	C	C4	...
39703	94838	73673	3000	3000	2550.00000	36 months	10.28%	97.20	C	C1	...
39704	93277	93254	3000	3000	950.00000	36 months	8.70%	94.98	B	B1	...
39705	93061	93057	5000	5000	250.00000	36 months	7.43%	155.38	A	A2	...
39706	92676	92671	5000	5000	150.00000	36 months	8.07%	156.84	A	A4	...
39707	92666	92661	5000	5000	525.00000	36 months	9.33%	159.77	B	B3	...
39708	92552	92542	5000	5000	375.00000	36 months	9.96%	161.25	B	B5	...
39709	92533	92529	5000	5000	675.00000	36 months	11.22%	164.23	C	C4	...
39710	92507	92502	5000	5000	250.00000	36 months	7.43%	155.38	A	A2	...
39711	92402	92390	5000	5000	700.00000	36 months	8.70%	158.30	B	B1	...
39712	92187	92174	2500	2500	1075.00000	36 months	8.07%	78.42	A	A4	...
39713	90665	90607	8500	8500	875.00000	36 months	10.28%	275.38	C	C1	...
39714	90395	90390	5000	5000	1325.00000	36 months	8.07%	156.84	A	A4	...
39715	90376	89243	5000	5000	650.00000	36 months	7.43%	155.38	A	A2	...
39716	87023	86999	7500	7500	800.00000	36 months	13.75%	255.43	E	E2	...

39717 rows × 111 columns

In [7]:

```
#check datatypes
loan_new.dtypes
```

Out[7]:

```
id                              int64
```

```
id                          int64
member_id                   int64
loan_amnt                   int64
funded_amnt                 int64
funded_amnt_inv           float64
term                       object
int_rate                   object
installment               float64
grade                      object
sub_grade                  object
emp_title                  object
emp_length                 object
home_ownership             object
annual_inc                float64
verification_status        object
issue_d                    object
loan_status                object
url                        object
desc                       object
purpose                    object
title                      object
zip_code                   object
addr_state                 object
dti                       float64
delinq_2yrs                 int64
earliest_cr_line           object
inq_last_6mths              int64
open_acc                    int64
pub_rec                     int64
revol_bal                   int64
revol_util                 object
total_acc                   int64
total_pymnt               float64
total_pymnt_inv           float64
total_rec_prncp           float64
total_rec_int             float64
total_rec_late_fee        float64
recoveries                float64
collection_recovery_fee   float64
last_pymnt_d               object
last_pymnt_amnt           float64
last_credit_pull_d         object
pub_rec_bankruptcies      float64
dtype: object
```

> Data Analysis

In [8]:

```
#To create Customer Behaviour Variable at the time of loan application
b_var =  [
 "delinq_2yrs",
  "earliest_cr_line",
  "inq_last_6mths",
  "open_acc",
  "pub_rec",
  "revol_bal",
  "revol_util",
  "total_acc",
  "out_prncp",
  "out_prncp_inv",
  "total_pymnt",
  "total_pymnt_inv",
  "total_rec_prncp",
  "total_rec_int",
  "total_rec_late_fee",
  "recoveries",
  "collection_recovery_fee",
  "last_pymnt_d",
  "last_pymnt_amnt",
  "last_credit_pull_d",
  "application_type"]
b_var
```

Out[8]:

```
['delinq_2yrs',
 'earliest_cr_line',
 'inq_last_6mths',
 'open_acc',
 'pub_rec',
 'revol_bal',
 'revol_util',
 'total_acc',
 'out_prncp',
 'out_prncp_inv',
 'total_pymnt',
 'total_pymnt_inv',
 'total_rec_prncp',
 'total_rec_int',
 'total_rec_late_fee',
 'recoveries',
 'collection_recovery_fee',
 'last_pymnt_d',
 'last_pymnt_amnt',
 'last_credit_pull_d',
 'application_type']
```

In [9]:

```
# let's now remove the behaviour variables from analysis
df =loan_file.drop(b_var, axis=1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Data columns (total 90 columns):
id                                  39717 non-null int64
member_id                           39717 non-null int64
loan_amnt                           39717 non-null int64
funded_amnt                         39717 non-null int64
funded_amnt_inv                     39717 non-null float64
term                                39717 non-null object
int_rate                            39717 non-null object
installment                         39717 non-null float64
grade                               39717 non-null object
sub_grade                           39717 non-null object
emp_title                           37258 non-null object
emp_length                          38642 non-null object
home_ownership                      39717 non-null object
annual_inc                          39717 non-null float64
verification_status                 39717 non-null object
issue_d                             39717 non-null object
loan_status                         39717 non-null object
pymnt_plan                          39717 non-null object
url                                 39717 non-null object
desc                                26777 non-null object
purpose                             39717 non-null object
title                               39706 non-null object
zip_code                            39717 non-null object
addr_state                          39717 non-null object
dti                                 39717 non-null float64
mths_since_last_delinq              14035 non-null float64
mths_since_last_record              2786 non-null float64
initial_list_status                 39717 non-null object
next_pymnt_d                        1140 non-null object
collections_12_mths_ex_med          39661 non-null float64
mths_since_last_major_derog         0 non-null float64
policy_code                         39717 non-null int64
annual_inc_joint                    0 non-null float64
dti_joint                           0 non-null float64
verification_status_joint           0 non-null float64
acc_now_delinq                      39717 non-null int64
tot_coll_amt                        0 non-null float64
tot_cur_bal                         0 non-null float64
open_acc_6m                         0 non-null float64
open_il_6m                          0 non-null float64
open_il_12m                         0 non-null float64
open_il_24m                         0 non-null float64
mths_since_rcnt_il                  0 non-null float64
total_bal_il                        0 non-null float64
```

```
total_bal_il                      0 non-null float64
il_util                           0 non-null float64
open_rv_12m                       0 non-null float64
open_rv_24m                       0 non-null float64
max_bal_bc                        0 non-null float64
all_util                          0 non-null float64
total_rev_hi_lim                  0 non-null float64
inq_fi                            0 non-null float64
total_cu_tl                       0 non-null float64
inq_last_12m                      0 non-null float64
acc_open_past_24mths              0 non-null float64
avg_cur_bal                       0 non-null float64
bc_open_to_buy                    0 non-null float64
bc_util                           0 non-null float64
chargeoff_within_12_mths          39661 non-null float64
delinq_amnt                       39717 non-null int64
mo_sin_old_il_acct                0 non-null float64
mo_sin_old_rev_tl_op              0 non-null float64
mo_sin_rcnt_rev_tl_op             0 non-null float64
mo_sin_rcnt_tl                    0 non-null float64
mort_acc                          0 non-null float64
mths_since_recent_bc              0 non-null float64
mths_since_recent_bc_dlq          0 non-null float64
mths_since_recent_inq             0 non-null float64
mths_since_recent_revol_delinq    0 non-null float64
num_accts_ever_120_pd             0 non-null float64
num_actv_bc_tl                    0 non-null float64
num_actv_rev_tl                   0 non-null float64
num_bc_sats                       0 non-null float64
num_bc_tl                         0 non-null float64
num_il_tl                         0 non-null float64
num_op_rev_tl                     0 non-null float64
num_rev_accts                     0 non-null float64
num_rev_tl_bal_gt_0               0 non-null float64
num_sats                          0 non-null float64
num_tl_120dpd_2m                  0 non-null float64
num_tl_30dpd                      0 non-null float64
num_tl_90g_dpd_24m                0 non-null float64
num_tl_op_past_12m                0 non-null float64
pct_tl_nvr_dlq                    0 non-null float64
percent_bc_gt_75                  0 non-null float64
pub_rec_bankruptcies              39020 non-null float64
tax_liens                         39678 non-null float64
tot_hi_cred_lim                   0 non-null float64
total_bal_ex_mort                 0 non-null float64
total_bc_limit                    0 non-null float64
total_il_high_credit_limit        0 non-null float64
dtypes: float64(64), int64(7), object(19)
memory usage: 27.3+ MB
```

In [10]:

```
# removing the columns having more than 90% missing values
missing_columns = loan_file.columns[100*(loan_file.isnull().sum()/len(loan_file.index)) > 90]
print(missing_columns)
```

```
Index(['mths_since_last_record', 'next_pymnt_d', 'mths_since_last_major_derog',
       'annual_inc_joint', 'dti_joint', 'verification_status_joint',
       'tot_coll_amt', 'tot_cur_bal', 'open_acc_6m', 'open_il_6m',
       'open_il_12m', 'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il',
       'il_util', 'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util',
       'total_rev_hi_lim', 'inq_fi', 'total_cu_tl', 'inq_last_12m',
       'acc_open_past_24mths', 'avg_cur_bal', 'bc_open_to_buy', 'bc_util',
       'mo_sin_old_il_acct', 'mo_sin_old_rev_tl_op', 'mo_sin_rcnt_rev_tl_op',
       'mo_sin_rcnt_tl', 'mort_acc', 'mths_since_recent_bc',
       'mths_since_recent_bc_dlq', 'mths_since_recent_inq',
       'mths_since_recent_revol_delinq', 'num_accts_ever_120_pd',
       'num_actv_bc_tl', 'num_actv_rev_tl', 'num_bc_sats', 'num_bc_tl',
       'num_il_tl', 'num_op_rev_tl', 'num_rev_accts', 'num_rev_tl_bal_gt_0',
       'num_sats', 'num_tl_120dpd_2m', 'num_tl_30dpd', 'num_tl_90g_dpd_24m',
       'num_tl_op_past_12m', 'pct_tl_nvr_dlq', 'percent_bc_gt_75',
       'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
       'total_il_high_credit_limit'],
      dtype='object')
```

In [11]:

```
loan = loan_file.drop(missing_columns, axis=1)
print(loan.shape)
```

```
(39717, 55)
```

In [12]:

```
# summarise number of missing values again
Sumr=100*(loan.isnull().sum()/len(loan.index))
print(Sumr)
```

```
id                             0.000000
member_id                      0.000000
loan_amnt                      0.000000
funded_amnt                    0.000000
funded_amnt_inv                0.000000
term                           0.000000
int_rate                       0.000000
installment                    0.000000
grade                          0.000000
sub_grade                      0.000000
emp_title                      6.191303
emp_length                     2.706650
home_ownership                 0.000000
annual_inc                     0.000000
verification_status            0.000000
issue_d                        0.000000
loan_status                    0.000000
pymnt_plan                     0.000000
url                            0.000000
desc                          32.580507
purpose                        0.000000
title                          0.027696
zip_code                       0.000000
addr_state                     0.000000
dti                            0.000000
delinq_2yrs                    0.000000
earliest_cr_line               0.000000
inq_last_6mths                 0.000000
mths_since_last_delinq        64.662487
open_acc                       0.000000
pub_rec                        0.000000
revol_bal                      0.000000
revol_util                     0.125891
total_acc                      0.000000
initial_list_status            0.000000
out_prncp                      0.000000
out_prncp_inv                  0.000000
total_pymnt                    0.000000
total_pymnt_inv                0.000000
total_rec_prncp                0.000000
total_rec_int                  0.000000
total_rec_late_fee             0.000000
recoveries                     0.000000
collection_recovery_fee        0.000000
last_pymnt_d                   0.178765
last_pymnt_amnt                0.000000
last_credit_pull_d             0.005036
collections_12_mths_ex_med     0.140998
policy_code                    0.000000
application_type               0.000000
acc_now_delinq                 0.000000
chargeoff_within_12_mths       0.140998
delinq_amnt                    0.000000
pub_rec_bankruptcies           1.754916
tax_liens                      0.098195
dtype: float64
```

In [13]:

```
# missing values in rows
```

```
loan_file.isnull().sum(axis=1)
```

Out[13]:

```
0        58
1        57
2        59
3        56
4        55
5        58
6        57
7        57
8        58
9        57
10       57
11       58
12       57
13       57
14       58
15       58
16       57
17       57
18       56
19       58
20       57
21       57
22       57
23       58
24       58
25       58
26       58
27       56
28       57
29       57
         ..
39687    59
39688    61
39689    59
39690    59
39691    59
39692    60
39693    59
39694    59
39695    59
39696    59
39697    59
39698    59
39699    59
39700    60
39701    59
39702    59
39703    59
39704    60
39705    59
39706    60
39707    59
39708    59
39709    60
39710    60
39711    59
39712    59
39713    59
39714    61
39715    61
39716    59
Length: 39717, dtype: int64
```

In [14]:

```
loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Data columns (total 55 columns):
id                            39717 non-null int64
```

```
member_id                    39717 non-null int64
loan_amnt                    39717 non-null int64
funded_amnt                  39717 non-null int64
funded_amnt_inv              39717 non-null float64
term                         39717 non-null object
int_rate                     39717 non-null object
installment                  39717 non-null float64
grade                        39717 non-null object
sub_grade                    39717 non-null object
emp_title                    37258 non-null object
emp_length                   38642 non-null object
home_ownership               39717 non-null object
annual_inc                   39717 non-null float64
verification_status          39717 non-null object
issue_d                      39717 non-null object
loan_status                  39717 non-null object
pymnt_plan                   39717 non-null object
url                          39717 non-null object
desc                         26777 non-null object
purpose                      39717 non-null object
title                        39706 non-null object
zip_code                     39717 non-null object
addr_state                   39717 non-null object
dti                          39717 non-null float64
delinq_2yrs                  39717 non-null int64
earliest_cr_line             39717 non-null object
inq_last_6mths               39717 non-null int64
mths_since_last_delinq       14035 non-null float64
open_acc                     39717 non-null int64
pub_rec                      39717 non-null int64
revol_bal                    39717 non-null int64
revol_util                   39667 non-null object
total_acc                    39717 non-null int64
initial_list_status          39717 non-null object
out_prncp                    39717 non-null float64
out_prncp_inv                39717 non-null float64
total_pymnt                  39717 non-null float64
total_pymnt_inv              39717 non-null float64
total_rec_prncp              39717 non-null float64
total_rec_int                39717 non-null float64
total_rec_late_fee           39717 non-null float64
recoveries                   39717 non-null float64
collection_recovery_fee      39717 non-null float64
last_pymnt_d                 39646 non-null object
last_pymnt_amnt              39717 non-null float64
last_credit_pull_d           39715 non-null object
collections_12_mths_ex_med   39661 non-null float64
policy_code                  39717 non-null int64
application_type             39717 non-null object
acc_now_delinq               39717 non-null int64
chargeoff_within_12_mths     39661 non-null float64
delinq_amnt                  39717 non-null int64
pub_rec_bankruptcies         39020 non-null float64
tax_liens                    39678 non-null float64
dtypes: float64(19), int64(13), object(23)
memory usage: 16.7+ MB
```

In [15]:

```
#we will not be able to use the variables zip code, address, state etc.
df = df.drop(['title', 'url', 'zip_code', 'addr_state'], axis=1)
```

In [16]:

```
df['loan_status'] = df['loan_status'].astype('category')
df['loan_status'].value_counts()
```

Out[16]:

```
Fully Paid     32950
Charged Off     5627
Current         1140
Name: loan_status, dtype: int64
```

In [17]:

```
# filtering only fully paid or charged-off
df = df[df['loan_status'] != 'Current']
df['loan_status'] = df['loan_status'].apply(lambda x: 0 if x=='Fully Paid' else 1)
```

In [18]:

```
# converting loan_status to integer type
df['loan_status'] = df['loan_status'].apply(lambda x: pd.to_numeric(x))
```

In [19]:

```
# summarising the values
df['loan_status'].value_counts()
```

Out[19]:

```
0    32950
1     5627
Name: loan_status, dtype: int64
```

> UNIVARIATE ANALYSIS

In [20]:

```
# default rate
round(np.mean(df['loan_status']), 2)
```

Out[20]:

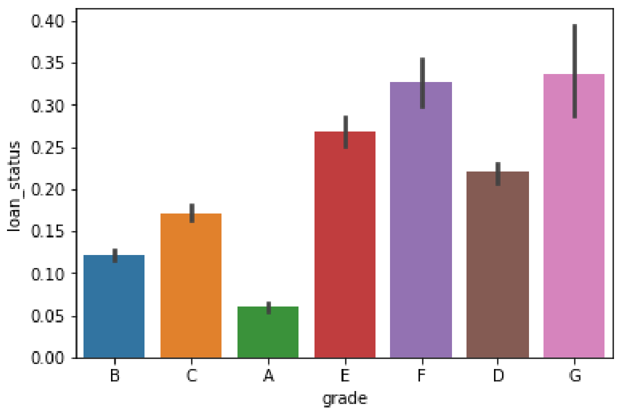
```
0.15
```

In [21]:

```
# plotting graphs

import matplotlib.pyplot as plt
import seaborn as sns
```

In [22]:

```
# plotting default rates across grade of the loan
sns.barplot(x='grade', y='loan_status', data=df)
plt.show()
```

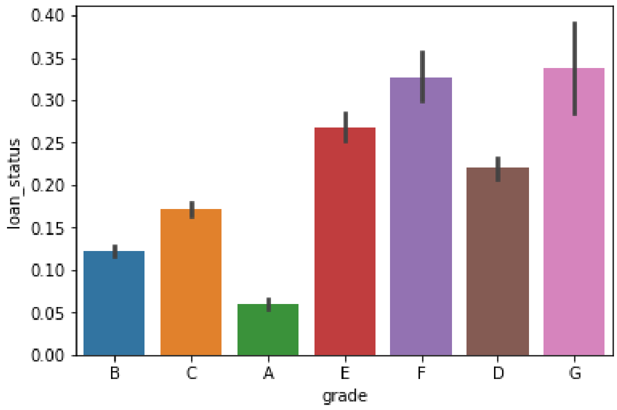


In [23]:

```
# lets define a function to plot loan_status across categorical variables
def plot_cat(cat_var):
    sns.barplot(x=cat_var, y='loan_status', data=df)
    plt.show()
```

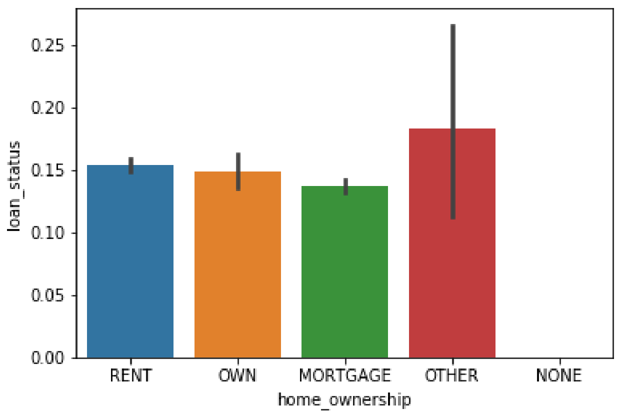
In [24]:

```
# compare default rates across grade of loan
#Clearly, as the grade of loan goes from A to G, the default rate increases. This is expected beca
use the grade is decided by Lending Club based on the riskiness of the loan.
plot_cat('grade')
```



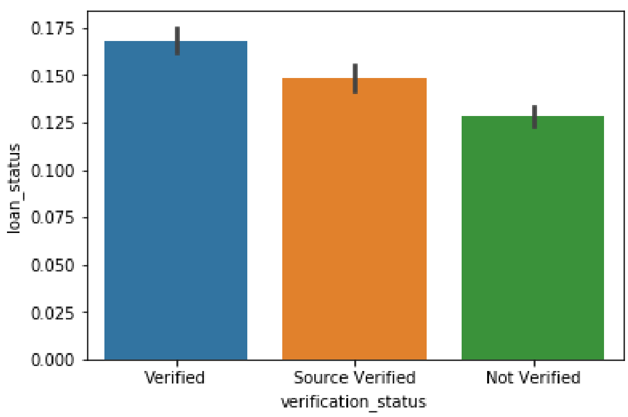
In [25]:

```
# home ownership: not a great discriminator
plot_cat('home_ownership')
```



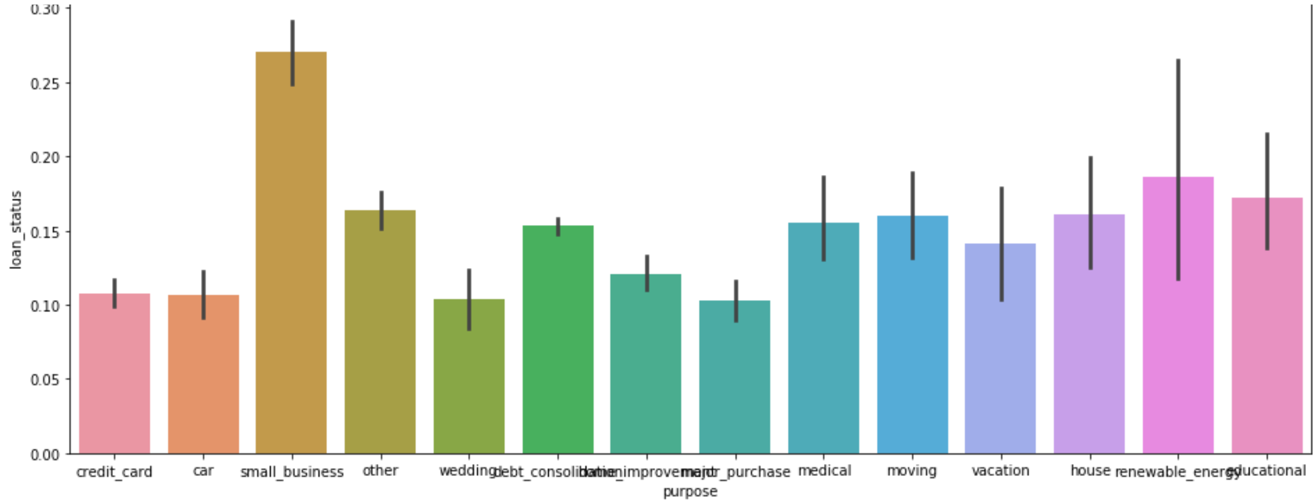
In [26]:

```
# verification_status: surprisingly, verified loans default more than not verifiedb
plot_cat('verification_status')
```



In [27]:

```
# purpose: small business loans defualt the most, then renewable energy and education
plt.figure(figsize=(16, 6))
plot_cat('purpose')
```



In [28]:

```
# let's also observe the distribution of loans across years
# first lets convert the year column into datetime and then extract year and month from it
df['issue_d'].head()
```

Out[28]:

```
0    Dec-11
1    Dec-11
2    Dec-11
3    Dec-11
5    Dec-11
Name: issue_d, dtype: object
```

In [29]:

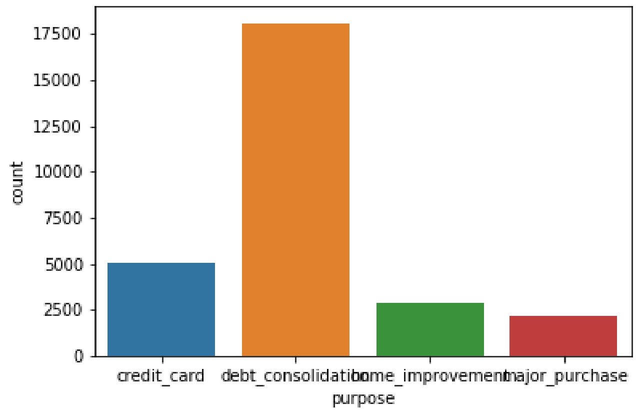
```
# filtering the df for the 4 types of loans mentioned above
main_purposes = ["credit_card","debt_consolidation","home_improvement","major_purchase"]
df = df[df['purpose'].isin(main_purposes)]
df['purpose'].value_counts()
```

Out[29]:

```
debt_consolidation    18055
credit_card            5027
home_improvement       2875
major_purchase         2150
Name: purpose, dtype: int64
```

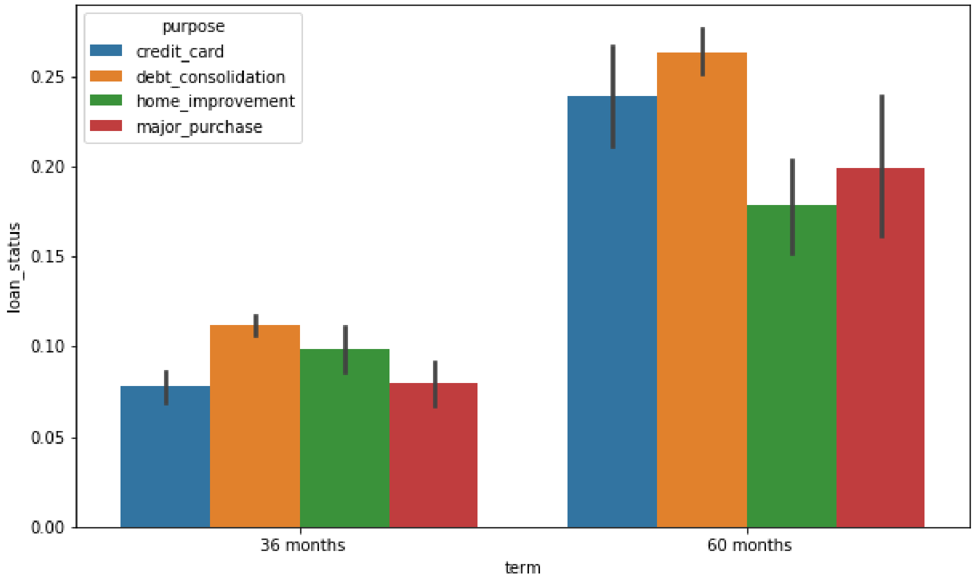
In [30]:

```
# plotting number of loans by purpose
sns.countplot(x=df['purpose'])
plt.show()
```



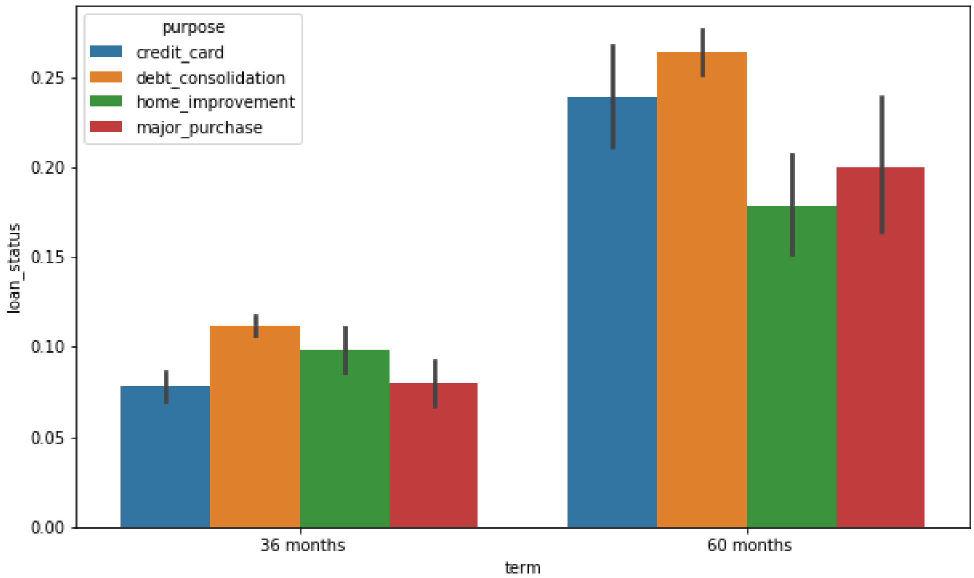
In [31]:

```python
# let's now compare the default rates across two types of categorical variables
# purpose of loan (constant) and another categorical variable (which changes)

plt.figure(figsize=[10, 6])
sns.barplot(x='term', y="loan_status", hue='purpose', data=df)
plt.show()
```



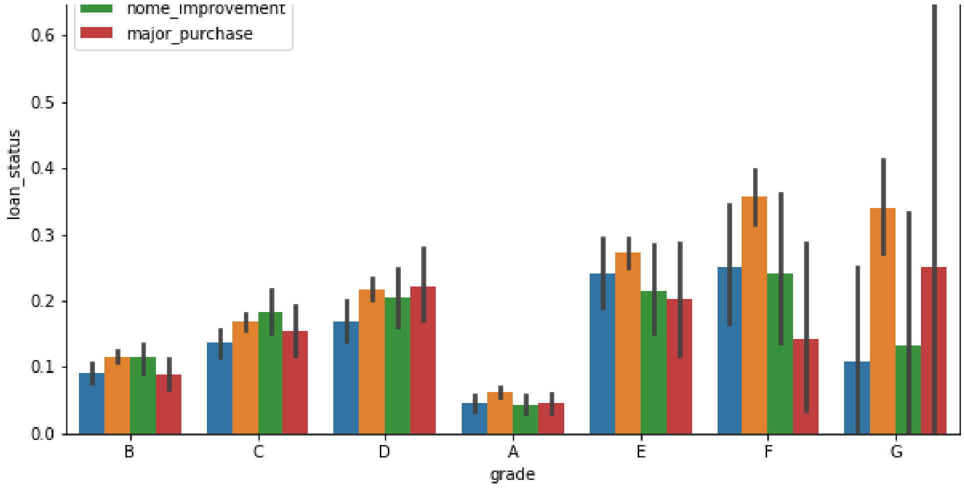
In [32]:

```python
# lets write a function which takes a categorical variable and plots the default rate
# segmented by purpose

def plot_segmented(cat_var):
    plt.figure(figsize=(10, 6))
    sns.barplot(x=cat_var, y='loan_status', hue='purpose', data=df)
    plt.show()


plot_segmented('term')
```



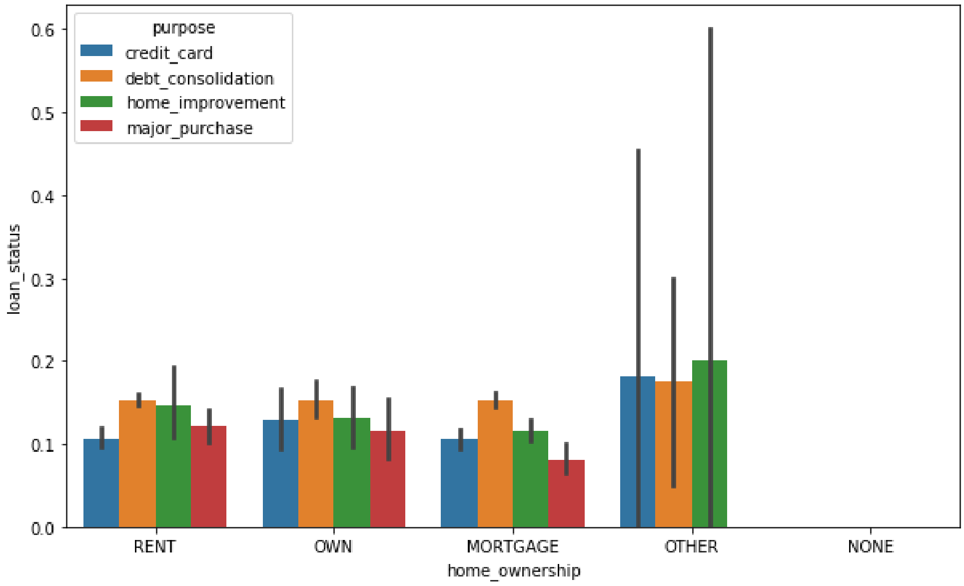
In [33]:

```python
# grade of loan
plot_segmented('grade')
```



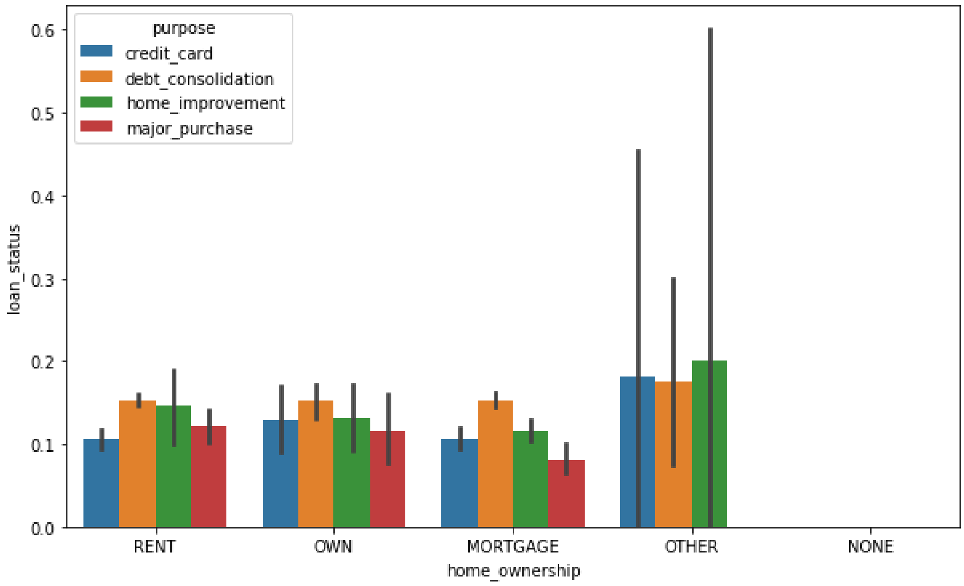
In [34]:

```
# home ownership
plot_segmented('home_ownership')
```



In [35]:

```
# home ownership
plot_segmented('home_ownership')
```



In [36]:

```
# variation of default rate across annual_inc
df.groupby('annual_inc').loan_status.mean().sort_values(ascending=False)
```

Out[36]:

```
annual_inc
4080.00      1.0
31008.00     1.0
49580.00     1.0
30548.00     1.0
30660.00     1.0
30696.00     1.0
103050.00    1.0
49439.00     1.0
62550.00     1.0
30992.00     1.0
101837.28    1.0
31504.27     1.0
62664.00     1.0
101657.00    1.0
62695.00     1.0
62742.00     1.0
31323.00     1.0
31356.00     1.0
42480.00     1.0
100650.00    1.0
49590.00     1.0
104085.00    1.0
30480.00     1.0
49632.00     1.0
61700.00     1.0
29744.00     1.0
29784.00     1.0
29808.00     1.0
29856.00     1.0
29865.00     1.0
            ...
70404.00     0.0
71219.20     0.0
71280.00     0.0
72204.00     0.0
71935.00     0.0
72194.00     0.0
72174.00     0.0
72150.00     0.0
72136.00     0.0
72100.00     0.0
72096.00     0.0
72072.00     0.0
72060.00     0.0
72054.00     0.0
71991.00     0.0
71964.00     0.0
71887.00     0.0
71352.00     0.0
71874.00     0.0
71820.00     0.0
71738.00     0.0
71711.00     0.0
71700.00     0.0
71688.00     0.0
71499.00     0.0
71496.00     0.0
71480.00     0.0
71400.00     0.0
71376.00     0.0
59617.48     0.0
Name: loan_status, Length: 4079, dtype: float64
```

> CoRelation Map

In [37]:

```
k =loan_new.select_dtypes(include=[np.number]).columns.size
correlation = loan_new.select_dtypes(include=[np.number]).corr()
correlation
```

Out[37]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	installment	annual_inc	
id	1.000000	0.993534	0.120614	0.131283	0.231603	0.076088	0.005572	0.0917
member_id	0.993534	1.000000	0.120393	0.130307	0.241324	0.070918	0.006442	0.0929
loan_amnt	0.120614	0.120393	1.000000	0.981790	0.937922	0.932260	0.268999	0.0624
funded_amnt	0.131283	0.130307	0.981790	1.000000	0.956172	0.958035	0.264798	0.0621
funded_amnt_inv	0.231603	0.241324	0.937922	0.956172	1.000000	0.905464	0.251981	0.0706
installment	0.076088	0.070918	0.932260	0.958035	0.905464	1.000000	0.267842	0.0520
annual_inc	0.005572	0.006442	0.268999	0.264798	0.251981	0.267842	1.000000	-0.1215
dti	0.091785	0.092910	0.062436	0.062194	0.070663	0.052038	-0.121530	1.0000
delinq_2yrs	-0.008417	-0.007905	-0.031951	-0.031866	-0.038171	-0.019755	0.022229	-0.0333
inq_last_6mths	-0.041021	-0.045879	0.012940	0.012857	-0.002800	0.011014	0.035465	0.0021
open_acc	0.016256	0.013804	0.177200	0.175682	0.162738	0.172893	0.156927	0.2878
pub_rec	-0.017683	-0.017066	-0.049997	-0.050576	-0.051470	-0.045706	-0.017864	-0.0047
revol_bal	0.001357	-0.001983	0.314022	0.306501	0.286265	0.309501	0.277374	0.2280
total_acc	0.039902	0.042217	0.256179	0.250551	0.242715	0.229860	0.234534	0.2291
total_pymnt	0.110432	0.111810	0.881910	0.898709	0.874730	0.858493	0.256313	0.0592
total_pymnt_inv	0.194832	0.205195	0.847635	0.864501	0.909127	0.817665	0.245198	0.0662
total_rec_prncp	0.092979	0.093773	0.845870	0.864082	0.838587	0.847762	0.256848	0.0367
total_rec_int	0.123268	0.126660	0.728343	0.736654	0.726736	0.642655	0.185056	0.1031
total_rec_late_fee	-0.055789	-0.058497	0.047103	0.049465	0.029379	0.058387	0.006814	-0.0114
recoveries	0.038686	0.036526	0.142789	0.143452	0.130997	0.121463	0.022184	0.0261
collection_recovery_fee	-0.010916	-0.012831	0.077005	0.078769	0.064282	0.077519	0.015981	0.0117
last_pymnt_amnt	0.142251	0.142582	0.474614	0.478448	0.469166	0.413588	0.143242	0.0085
pub_rec_bankruptcies	-0.007997	-0.007346	-0.035981	-0.036995	-0.041193	-0.033361	-0.016224	0.0059

23 rows × 23 columns