In [58]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [59]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [61]:

```
df=pd.read_csv(r'C:\Users\Lenovo\Downloads\lending_club_loans.csv',header=1)

xy=pd.read_csv(r'C:\Users\Lenovo\Documents\LCDataDictionary.csv')
```

In [62]:

```
pd.set_option("display.max_columns", None)
```

In [63]:

```
df.shape
```

Out[63]:

```
(42542, 115)
```

In [64]:

```
1 xy[:60]
```

Out[64]:

	LoanStatNew	Description
0	acc_now_delinq	The number of accounts on which the borrower i...
1	acc_open_past_24mths	Number of trades opened in past 24 months.
2	addr_state	The state provided by the borrower in the loan...
3	all_util	Balance to credit limit on all trades
4	annual_inc	The self-reported annual income provided by th...
5	annual_inc_joint	The combined self-reported annual income provi...
6	application_type	Indicates whether the loan is an individual ap...
7	avg_cur_bal	Average current balance of all accounts
8	bc_open_to_buy	Total open to buy on revolving bankcards.
9	bc_util	Ratio of total current balance to high credit/...
10	chargeoff_within_12_mths	Number of charge-offs within 12 months
11	collection_recovery_fee	post charge off collection fee
12	collections_12_mths_ex_med	Number of collections in 12 months excluding m...
13	delinq_2yrs	The number of 30+ days past-due incidences of ...
14	delinq_amnt	The past-due amount owed for the accounts on w...
15	desc	Loan description provided by the borrower
16	dti	A ratio calculated using the borrower’s total ...
17	dti_joint	A ratio calculated using the co-borrowers' tot...
18	earliest_cr_line	The month the borrower's earliest reported cre...
19	emp_length	Employment length in years. Possible values ar...
20	emp_title	The job title supplied by the Borrower when ap...
21	fico_range_high	The upper boundary range the borrower’s FICO a...
22	fico_range_low	The lower boundary range the borrower’s FICO a...
23	funded_amnt	The total amount committed to that loan at tha...
24	funded_amnt_inv	The total amount committed by investors for th...
25	grade	LC assigned loan grade
26	home_ownership	The home ownership status provided by the borr...
27	id	A unique LC assigned ID for the loan listing.
28	il_util	Ratio of total current balance to high credit/...
29	initial_list_status	The initial listing status of the loan. Possib...
30	inq_fi	Number of personal finance inquiries
31	inq_last_12m	Number of credit inquiries in past 12 months
32	inq_last_6mths	The number of inquiries in past 6 months (excl...
33	installment	The monthly payment owed by the borrower if th...

	LoanStatNew	Description
34	int_rate	Interest Rate on the loan
35	issue_d	The month which the loan was funded
36	last_credit_pull_d	The most recent month LC pulled credit for thi...
37	last_fico_range_high	The upper boundary range the borrower’s last F...
38	last_fico_range_low	The lower boundary range the borrower’s last F...
39	last_pymnt_amnt	Last total payment amount received
40	last_pymnt_d	Last month payment was received
41	loan_amnt	The listed amount of the loan applied for by t...
42	loan_status	Current status of the loan
43	max_bal_bc	Maximum current balance owed on all revolving ...
44	member_id	A unique LC assigned Id for the borrower member.
45	mo_sin_old_il_acct	Months since oldest bank installment account o...
46	mo_sin_old_rev_tl_op	Months since oldest revolving account opened
47	mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
48	mo_sin_rcnt_tl	Months since most recent account opened
49	mort_acc	Number of mortgage accounts.
50	mths_since_last_delinq	The number of months since the borrower's last...
51	mths_since_last_major_derog	Months since most recent 90-day or worse rating
52	mths_since_last_record	The number of months since the last public rec...
53	mths_since_rcnt_il	Months since most recent installment accounts ...
54	mths_since_recent_bc	Months since most recent bankcard account opened.
55	mths_since_recent_bc_dlq	Months since most recent bankcard delinquency
56	mths_since_recent_inq	Months since most recent inquiry.
57	mths_since_recent_revol_delinq	Months since most recent revolving delinquency.
58	next_pymnt_d	Next scheduled payment date
59	num_accts_ever_120_pd	Number of accounts ever 120 or more days past due

In [65]:

```
1 xy[60:]
```

Out[65]:

	LoanStatNew	Description
60	num_actv_bc_tl	Number of currently active bankcard accounts
61	num_actv_rev_tl	Number of currently active revolving trades
62	num_bc_sats	Number of satisfactory bankcard accounts
63	num_bc_tl	Number of bankcard accounts
64	num_il_tl	Number of installment accounts
65	num_op_rev_tl	Number of open revolving accounts
66	num_rev_accts	Number of revolving accounts
67	num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
68	num_sats	Number of satisfactory accounts
69	num_tl_120dpd_2m	Number of accounts currently 120 days past due...
70	num_tl_30dpd	Number of accounts currently 30 days past due ...
71	num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in...
72	num_tl_op_past_12m	Number of accounts opened in past 12 months
73	open_acc	The number of open credit lines in the borrowe...
74	open_acc_6m	Number of open trades in last 6 months
75	open_il_12m	Number of installment accounts opened in past ...
76	open_il_24m	Number of installment accounts opened in past ...
77	open_il_6m	Number of currently active installment trades
78	open_rv_12m	Number of revolving trades opened in past 12 m...
79	open_rv_24m	Number of revolving trades opened in past 24 m...
80	out_prncp	Remaining outstanding principal for total amou...
81	out_prncp_inv	Remaining outstanding principal for portion of...
82	pct_tl_nvr_dlq	Percent of trades never delinquent
83	percent_bc_gt_75	Percentage of all bankcard accounts > 75% of l...
84	policy_code	publicly available policy_code=1\nnew products...
85	pub_rec	Number of derogatory public records
86	pub_rec_bankruptcies	Number of public record bankruptcies
87	purpose	A category provided by the borrower for the lo...
88	pymnt_plan	Indicates if a payment plan has been put in pl...
89	recoveries	post charge off gross recovery
90	revol_bal	Total credit revolving balance
91	revol_util	Revolving line utilization rate, or the amount...
92	sub_grade	LC assigned loan subgrade
93	tax_liens	Number of tax liens

	LoanStatNew	Description
94	term	The number of payments on the loan. Values are...
95	title	The loan title provided by the borrower
96	tot_coll_amt	Total collection amounts ever owed
97	tot_cur_bal	Total current balance of all accounts
98	tot_hi_cred_lim	Total high credit/credit limit
99	total_acc	The total number of credit lines currently in ...
100	total_bal_ex_mort	Total credit balance excluding mortgage
101	total_bal_il	Total current balance of all installment accounts
102	total_bc_limit	Total bankcard high credit/credit limit
103	total_cu_tl	Number of finance trades
104	total_il_high_credit_limit	Total installment high credit/credit limit
105	total_pymnt	Payments received to date for total amount funded
106	total_pymnt_inv	Payments received to date for portion of total...
107	total_rec_int	Interest received to date
108	total_rec_late_fee	Late fees received to date
109	total_rec_prncp	Principal received to date
110	total_rev_hi_lim	Total revolving high credit/credit limit
111	url	URL for the LC page with listing data.
112	verification_status	Indicates if income was verified by LC, not ve...
113	verified_status_joint	Indicates if the co-borrowers' joint income wa...
114	zip_code	The first 3 numbers of the zip code provided b...
115	NaN	NaN
116	NaN	* Employer Title replaces Employer Name for al...

In [66]:

```
df.head(2)
```

Out[66]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installmer
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.8
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.8

In [67]:

```
# checking null values in each column & number of unique values

for i in df.columns.values:
    print(i,',',df[i].isnull().sum(),',',df[i].nunique())
```

```
id , 4 , 42538
member_id , 7 , 42535
loan_amnt , 7 , 898
funded_amnt , 7 , 1051
funded_amnt_inv , 7 , 9240
term , 7 , 2
int_rate , 7 , 394
installment , 7 , 16459
grade , 7 , 7
sub_grade , 7 , 35
emp_title , 2633 , 30658
emp_length , 1119 , 11
home_ownership , 7 , 5
annual_inc , 11 , 5597
verification_status , 7 , 3
issue_d , 7 , 55
loan_status , 7 , 9
pymnt_plan , 7 , 2
url , 7 , 42535
```

In [68]:

```
# Dropping all columns which are totally empty
df.dropna(how='all', axis=1, inplace=True)
```

In [69]:

```
df.shape
```

Out[69]:

(42542, 61)

In [70]:

```
df.head(2)
```

Out[70]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installmer
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.8
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.8

In [71]:

```
# Droping columns which consist of almost all values
for i in df.columns.values:
    if ((df[i].isnull().sum()/df.shape[0])>0.50):
        print(i,df[i].isnull().sum())
        df.drop(i, axis=1, inplace=True)
```

```
mths_since_last_delinq 26933
mths_since_last_record 38891
next_pymnt_d 39246
```

In [72]:

```
df.shape
```

Out[72]:

(42542, 58)

In [73]:

```
for i in df.columns.values:
    print(i,'-',df[i].isnull().sum(),',',df[i].nunique())
```

```
id - 4 , 42538
member_id - 7 , 42535
loan_amnt - 7 , 898
funded_amnt - 7 , 1051
funded_amnt_inv - 7 , 9240
term - 7 , 2
int_rate - 7 , 394
installment - 7 , 16459
grade - 7 , 7
sub_grade - 7 , 35
emp_title - 2633 , 30658
emp_length - 1119 , 11
home_ownership - 7 , 5
annual_inc - 11 , 5597
verification_status - 7 , 3
issue_d - 7 , 55
loan_status - 7 , 9
pymnt_plan - 7 , 2
url - 7 , 42535
desc - 13300 , 28963
purpose - 7 , 14
title - 20 , 21253
zip_code - 7 , 837
addr_state - 7 , 50
dti - 7 , 2894
delinq_2yrs - 36 , 12
earliest_cr_line - 36 , 530
fico_range_low - 7 , 44
fico_range_high - 7 , 44
inq_last_6mths - 36 , 28
open_acc - 36 , 44
pub_rec - 36 , 6
revol_bal - 7 , 22709
revol_util - 97 , 1119
total_acc - 36 , 83
initial_list_status - 7 , 1
out_prncp - 7 , 547
out_prncp_inv - 7 , 548
total_pymnt - 7 , 40579
total_pymnt_inv - 7 , 40108
total_rec_prncp - 7 , 8214
total_rec_int - 7 , 37533
total_rec_late_fee - 7 , 1562
recoveries - 7 , 4530
collection_recovery_fee - 7 , 2857
last_pymnt_d - 90 , 106
last_pymnt_amnt - 7 , 37117
last_credit_pull_d - 11 , 111
last_fico_range_high - 7 , 72
last_fico_range_low - 7 , 71
collections_12_mths_ex_med - 152 , 1
policy_code - 7 , 1
application_type - 7 , 1
acc_now_delinq - 36 , 2
chargeoff_within_12_mths - 152 , 1
```

```
delinq_amnt - 36 , 3
pub_rec_bankruptcies - 1372 , 3
tax_liens - 112 , 2
```

In [74]:

```
1 df['emp_title'].value_counts()
```

Out[74]:

```
US Army                             139
Bank of America                     115
IBM                                  72
Kaiser Permanente                    61
AT&T                                 61
                                   ...
Regional Elite Airlines Services      1
Mass General Medical Group            1
Kontera                               1
Southeast Georgia Health ystem        1
Homemaker                             1
Name: emp_title, Length: 30658, dtype: int64
```

In [75]:

```
1 df['emp_title'].fillna('Other', inplace=True)
```

In [76]:

```
1 df['emp_length'].value_counts()
```

Out[76]:

```
10+ years    9369
< 1 year     5062
2 years      4743
3 years      4364
4 years      3649
1 year       3595
5 years      3458
6 years      2375
7 years      1875
8 years      1592
9 years      1341
Name: emp_length, dtype: int64
```

In [77]:

```
1 df['emp_length'].fillna('< 1 year', inplace=True)
```

In [78]:

```
1 df.drop('url',axis=1,inplace=True)
```

In [79]:

```
df['desc'].value_counts()
```

Out[79]:

```
225
Debt Consolidation
11
Camping Membership
8
refinancing
5
personal loan
3

...
  Borrower added on 04/13/11 > debt consolidation<br/> Borrower added on 04/
13/11 > debt consolidation<br/>
1
  Borrower added on 04/13/11 > I am looking to pay the difference in the sel
l of my home that I have up for sale and in contract. I am looking at paying
$20000 of the difference so I can relocate closer to me and my wife's employ
ers. We have no other debt, to include credit cards, with the exception of a
n auto loan at $19000. I plan on solely paying this debt off and ultimately
purchasing a home again in a closer location. I initially needed a $12000 lo
an , but now I am only needing the requested amount. Thank you.<br/>      1
  Borrower added on 04/18/11 > I am employed for 23 years at a major aerospa
ce company.<br/>
1
I'm in the process of doing home remolding.
1
I need to make several improvements around the house - fix garage, fix back
fencing, and misc other.
1
Name: desc, Length: 28963, dtype: int64
```

In [80]:

```
df.drop('desc', axis=1, inplace=True)
```

In [81]:

```
df['pub_rec_bankruptcies'].value_counts()
```

Out[81]:

```
0.0    39316
1.0     1846
2.0        8
Name: pub_rec_bankruptcies, dtype: int64
```

In [82]:

```
df['pub_rec_bankruptcies'].fillna(0.0, inplace=True)
```

In [83]:

```
1 df.shape
```

Out[83]:

(42542, 56)

In [84]:

```
for i in df.columns.values:
    print(i,'-',df[i].isnull().sum())
```

```
id - 4
member_id - 7
loan_amnt - 7
funded_amnt - 7
funded_amnt_inv - 7
term - 7
int_rate - 7
installment - 7
grade - 7
sub_grade - 7
emp_title - 0
emp_length - 0
home_ownership - 7
annual_inc - 11
verification_status - 7
issue_d - 7
loan_status - 7
pymnt_plan - 7
purpose - 7
title - 20
zip_code - 7
addr_state - 7
dti - 7
delinq_2yrs - 36
earliest_cr_line - 36
fico_range_low - 7
fico_range_high - 7
inq_last_6mths - 36
open_acc - 36
pub_rec - 36
revol_bal - 7
revol_util - 97
total_acc - 36
initial_list_status - 7
out_prncp - 7
out_prncp_inv - 7
total_pymnt - 7
total_pymnt_inv - 7
total_rec_prncp - 7
total_rec_int - 7
total_rec_late_fee - 7
recoveries - 7
collection_recovery_fee - 7
last_pymnt_d - 90
last_pymnt_amnt - 7
last_credit_pull_d - 11
last_fico_range_high - 7
last_fico_range_low - 7
collections_12_mths_ex_med - 152
policy_code - 7
application_type - 7
acc_now_delinq - 36
chargeoff_within_12_mths - 152
delinq_amnt - 36
```

```
pub_rec_bankruptcies - 0
tax_liens - 112
```

In [85]:

```
for i in df.columns.values:
    if df[i].nunique()<=1:
        print(i)
        print(df[i].value_counts())
        print('----------------------------------------------------')
        df.drop(i,axis=1,inplace=True)

```

```
initial_list_status
f    42535
Name: initial_list_status, dtype: int64
----------------------------------------------------
collections_12_mths_ex_med
0.0    42390
Name: collections_12_mths_ex_med, dtype: int64
----------------------------------------------------
policy_code
1.0    42535
Name: policy_code, dtype: int64
----------------------------------------------------
application_type
INDIVIDUAL    42535
Name: application_type, dtype: int64
----------------------------------------------------
chargeoff_within_12_mths
0.0    42390
Name: chargeoff_within_12_mths, dtype: int64
----------------------------------------------------
```

In [86]:

```
for i in df.columns.values:
    if df[i].nunique()<=5:
        print(i)
        print(df[i].value_counts())
        print(df[i].isna().sum())
        print('--------------------------------------------------')
```

```
term
 36 months    31534
 60 months    11001
Name: term, dtype: int64
7
--------------------------------------------------
home_ownership
RENT        20181
MORTGAGE    18959
OWN          3251
OTHER         136
NONE            8
Name: home_ownership, dtype: int64
7
--------------------------------------------------
verification_status
Not Verified       18758
Verified           13471
Source Verified    10306
Name: verification_status, dtype: int64
7
--------------------------------------------------
pymnt_plan
n    42534
y        1
Name: pymnt_plan, dtype: int64
7
--------------------------------------------------
acc_now_delinq
0.0    42502
1.0        4
Name: acc_now_delinq, dtype: int64
36
--------------------------------------------------
delinq_amnt
0.0       42504
27.0          1
6053.0        1
Name: delinq_amnt, dtype: int64
36
--------------------------------------------------
pub_rec_bankruptcies
0.0    40688
1.0     1846
2.0        8
Name: pub_rec_bankruptcies, dtype: int64
0
--------------------------------------------------
tax_liens
0.0    42429
1.0        1
```

```
Name: tax_liens, dtype: int64
112
--------------------------------------------------
```

In [87]:

```
1 drop_col=['pymnt_plan','acc_now_delinq','delinq_amnt','tax_liens']
```

In [88]:

```
1 df.drop(columns=drop_col,axis=1,inplace=True)
```

In [89]:

```
1 df.shape
```

Out[89]:

```
(42542, 47)
```

In [ ]:

```
1
```

In [90]:

```
for i in df.columns.values:
    print(i+'=',df[i].isnull().sum())
```

```
id= 4
member_id= 7
loan_amnt= 7
funded_amnt= 7
funded_amnt_inv= 7
term= 7
int_rate= 7
installment= 7
grade= 7
sub_grade= 7
emp_title= 0
emp_length= 0
home_ownership= 7
annual_inc= 11
verification_status= 7
issue_d= 7
loan_status= 7
purpose= 7
title= 20
zip_code= 7
addr_state= 7
dti= 7
delinq_2yrs= 36
earliest_cr_line= 36
fico_range_low= 7
fico_range_high= 7
inq_last_6mths= 36
open_acc= 36
pub_rec= 36
revol_bal= 7
revol_util= 97
total_acc= 36
out_prncp= 7
out_prncp_inv= 7
total_pymnt= 7
total_pymnt_inv= 7
total_rec_prncp= 7
total_rec_int= 7
total_rec_late_fee= 7
recoveries= 7
collection_recovery_fee= 7
last_pymnt_d= 90
last_pymnt_amnt= 7
last_credit_pull_d= 11
last_fico_range_high= 7
last_fico_range_low= 7
pub_rec_bankruptcies= 0
```

In [91]:

```
df['last_pymnt_amnt'].value_counts()
```

Out[91]:

```
0.00       96
200.00     19
100.00     17
50.00      17
150.00     13
           ..
4210.88     1
548.44      1
1367.16     1
2674.24     1
156.39      1
Name: last_pymnt_amnt, Length: 37117, dtype: int64
```

In [92]:

```
df['last_pymnt_amnt'].fillna(0,inplace=True)
```

In [93]:

```
df['inq_last_6mths'].value_counts()
```

Out[93]:

```
0.0     19657
1.0     11247
2.0      5987
3.0      3182
4.0      1056
5.0       596
6.0       339
7.0       182
8.0       115
9.0        50
10.0       24
11.0       15
12.0       15
15.0        9
13.0        6
14.0        6
18.0        4
16.0        3
17.0        2
24.0        2
19.0        2
32.0        1
33.0        1
31.0        1
28.0        1
25.0        1
27.0        1
20.0        1
Name: inq_last_6mths, dtype: int64
```

In [94]:

```
df['inq_last_6mths'].fillna(0,inplace=True)
```

In [95]:

```
df['delinq_2yrs'].value_counts()
```

Out[95]:

```
0.0     37771
1.0      3595
2.0       771
3.0       244
4.0        72
5.0        27
6.0        13
7.0         6
8.0         3
11.0        2
9.0         1
13.0        1
Name: delinq_2yrs, dtype: int64
```

In [96]:

```
df['delinq_2yrs'].fillna(0,inplace=True)
```

In [97]:

```
df['pub_rec'].value_counts()
```

Out[97]:

```
0.0    40130
1.0     2298
2.0       64
3.0       11
4.0        2
5.0        1
Name: pub_rec, dtype: int64
```

In [98]:

```
df['pub_rec'].fillna(0,inplace=True)
```

In [99]:

```
df['revol_bal'].value_counts()
```

Out[99]:

```
0.0        1119
255.0        14
298.0        14
1.0          13
682.0        12
           ...
14170.0       1
43734.0       1
37778.0       1
59797.0       1
5251.0        1
Name: revol_bal, Length: 22709, dtype: int64
```

In [100]:

```
df['revol_bal'].fillna(0,inplace=True)
```

In [101]:

```
df['out_prncp'].value_counts()
```

Out[101]:

```
0.00       41988
466.88         2
1486.27        1
628.93         1
1602.99        1
           ...
815.87         1
1827.92        1
1086.45        1
2804.26        1
409.05         1
Name: out_prncp, Length: 547, dtype: int64
```

In [102]:

```
df['out_prncp'].fillna(0,inplace=True)
```

In [103]:

```
df['total_pymnt'].value_counts()
```

Out[103]:

```
11196.569430    26
0.000000        21
10956.775960    16
11784.232230    16
13148.137860    15
                ..
29914.869330     1
2125.415518      1
4830.896581      1
6790.732439      1
5619.762090      1
Name: total_pymnt, Length: 40579, dtype: int64
```

In [104]:

```
df['total_pymnt'].fillna(0,inplace=True)
```

In [105]:

```
df['total_pymnt_inv'].value_counts()
```

Out[105]:

```
0.00        299
6514.52      16
5478.39      14
13148.14     14
11196.57     12
           ...
14728.42      1
30541.64      1
4215.88       1
4307.33       1
258.82        1
Name: total_pymnt_inv, Length: 40108, dtype: int64
```

In [106]:

```
df['total_pymnt_inv'].fillna(0,inplace=True)
```

In [107]:

```
df['total_rec_int'].value_counts()
```

Out[107]:

```
0.00       83
1196.57    26
514.52     19
956.78     17
1784.23    17
           ..
2036.58     1
8830.36     1
3282.14     1
306.58      1
619.76      1
Name: total_rec_int, Length: 37533, dtype: int64
```

In [108]:

```
df['total_rec_int'].fillna(0,inplace=True)
```

In [109]:

```
df['total_rec_late_fee'].value_counts()
```

Out[109]:

```
0.000000      40148
15.000000       298
15.000000        76
30.000000        65
15.000000        49
               ...
38.895205         1
29.759757         1
29.974896         1
18.000000         1
120.000000        1
Name: total_rec_late_fee, Length: 1562, dtype: int64
```

In [110]:

```
df['total_rec_late_fee'].fillna(0,inplace=True)
```

In [111]:

```
1 df['recoveries'].value_counts()
```

Out[111]:

```
0.00       37789
13.93          4
10.40          4
11.29          4
10.66          3
           ...
1476.07        1
36.33          1
429.14         1
245.49         1
4897.92        1
Name: recoveries, Length: 4530, dtype: int64
```

In [112]:

```
1 df['recoveries'].fillna(0,inplace=True)
```

In [113]:

```
1 df['collection_recovery_fee'].value_counts()
```

Out[113]:

```
0.0000       38243
2.0000          15
1.2000          11
1.6000          10
0.8000          10
             ...
7.9300           1
63.8456          1
6.3200           1
268.2936         1
1714.2700        1
Name: collection_recovery_fee, Length: 2857, dtype: int64
```

In [114]:

```
1 df['collection_recovery_fee'].fillna(0,inplace=True)
```

In [115]:

```
df['home_ownership'].value_counts()
```

Out[115]:

```
RENT        20181
MORTGAGE    18959
OWN          3251
OTHER         136
NONE            8
Name: home_ownership, dtype: int64
```

In [116]:

```
df['home_ownership'].fillna('NONE',inplace=True)
```

In [117]:

```
df['verification_status'].value_counts()
```

Out[117]:

```
Not Verified       18758
Verified           13471
Source Verified    10306
Name: verification_status, dtype: int64
```

In [118]:

```
df['verification_status'].fillna('Not Verified',inplace=True)
```

In [119]:

```
df['purpose'].value_counts()
```

Out[119]:

```
debt_consolidation    19776
credit_card            5477
other                  4425
home_improvement       3199
major_purchase         2311
small_business         1992
car                    1615
wedding                1004
medical                 753
moving                  629
house                   426
educational             422
vacation                400
renewable_energy        106
Name: purpose, dtype: int64
```

In [120]:

```
df['purpose'].fillna('other',inplace=True)
```

In [121]:

```
df['dti'].value_counts()
```

Out[121]:

```
0.00     206
12.00     54
18.00     46
10.00     46
19.20     45
         ...
28.04      1
27.44      1
26.87      1
27.58      1
28.43      1
Name: dti, Length: 2894, dtype: int64
```

In [122]:

```
df['dti'].fillna(0.00,inplace=True)
```

In [123]:

```
for i in df.columns.values:
    print(i+'=',df[i].isnull().sum())
```

```
id= 4
member_id= 7
loan_amnt= 7
funded_amnt= 7
funded_amnt_inv= 7
term= 7
int_rate= 7
installment= 7
grade= 7
sub_grade= 7
emp_title= 0
emp_length= 0
home_ownership= 0
annual_inc= 11
verification_status= 0
issue_d= 7
loan_status= 7
purpose= 0
title= 20
zip_code= 7
addr_state= 7
dti= 0
delinq_2yrs= 0
earliest_cr_line= 36
fico_range_low= 7
fico_range_high= 7
inq_last_6mths= 0
open_acc= 36
pub_rec= 0
revol_bal= 0
revol_util= 97
total_acc= 36
out_prncp= 0
out_prncp_inv= 7
total_pymnt= 0
total_pymnt_inv= 0
total_rec_prncp= 7
total_rec_int= 0
total_rec_late_fee= 0
recoveries= 0
collection_recovery_fee= 0
last_pymnt_d= 90
last_pymnt_amnt= 0
last_credit_pull_d= 11
last_fico_range_high= 7
last_fico_range_low= 7
pub_rec_bankruptcies= 0
```

In [124]:

```
df.dropna( axis=0, inplace=True)
```

In [125]:

```
for i in df.columns.values:
    print(i+'=',df[i].isnull().sum())
```

```
id= 0
member_id= 0
loan_amnt= 0
funded_amnt= 0
funded_amnt_inv= 0
term= 0
int_rate= 0
installment= 0
grade= 0
sub_grade= 0
emp_title= 0
emp_length= 0
home_ownership= 0
annual_inc= 0
verification_status= 0
issue_d= 0
loan_status= 0
purpose= 0
title= 0
zip_code= 0
addr_state= 0
dti= 0
delinq_2yrs= 0
earliest_cr_line= 0
fico_range_low= 0
fico_range_high= 0
inq_last_6mths= 0
open_acc= 0
pub_rec= 0
revol_bal= 0
revol_util= 0
total_acc= 0
out_prncp= 0
out_prncp_inv= 0
total_pymnt= 0
total_pymnt_inv= 0
total_rec_prncp= 0
total_rec_int= 0
total_rec_late_fee= 0
recoveries= 0
collection_recovery_fee= 0
last_pymnt_d= 0
last_pymnt_amnt= 0
last_credit_pull_d= 0
last_fico_range_high= 0
last_fico_range_low= 0
pub_rec_bankruptcies= 0
```

In [126]:

```
1 df.shape
```

Out[126]:

```
(42350, 47)
```

In [127]:

```
1 df.head()
```

Out[127]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installmer
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.8
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.8
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.3
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.3
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.7

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```

In [ ]:

```
1
```