Project of Loan Club

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv(r'C:\\Users\\vinit\\OneDrive\\Documents\\Punit sir Project\\Le
```

In [3]:
```
data
```

Out[3]:

	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...
...	...	...
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...

117 rows × 2 columns

In [4]:
```
dataset = pd.read_csv(r'C:\\Users\\vinit\\OneDrive\\Documents\\Punit sir Project\
```

```
C:\Users\vinit\AppData\Local\Temp\ipykernel_9452\1744156391.py:1: DtypeWarning:
Columns (0,49) have mixed types. Specify dtype option on import or set low_memo
ry=False.
  dataset = pd.read_csv(r'C:\\Users\\vinit\\OneDrive\\Documents\\Punit sir Proj
ect\\Leading Market Hub\\lending_club_loans.csv',skiprows=1)
```

In [5]:
```
dataset
```

Out[5]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installı
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	16
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	5
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	8
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	33
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	6
...	...	...	...	...	...	...	...	
42533	72176	70868.0	2525.0	2525.0	225.0	36 months	9.33%	8
42534	71623	70735.0	6500.0	6500.0	0.0	36 months	8.38%	20
42535	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	15
42536	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	
42537	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	

42538 rows × 115 columns

In [6]:
```
dataset.columns
```

Out[6]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade',
       ...
       'num_tl_90g_dpd_24m', 'num_tl_op_past_12m', 'pct_tl_nvr_dlq',
       'percent_bc_gt_75', 'pub_rec_bankruptcies', 'tax_liens',
       'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
       'total_il_high_credit_limit'],
      dtype='object', length=115)
```

In [ ]:

In [7]:
```
df = dataset.copy()
```

In [8]:
```
df
```

Out[8]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installı
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	16
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	5
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	8
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	33
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	6
...	...	...	...	...	...	...	...	
42533	72176	70868.0	2525.0	2525.0	225.0	36 months	9.33%	8
42534	71623	70735.0	6500.0	6500.0	0.0	36 months	8.38%	20
42535	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	15
42536	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	
42537	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	

42538 rows × 115 columns

In [9]:
```
df.shape
```

Out[9]:
```
(42538, 115)
```

In [10]:
```
data.shape
```

Out[10]:
```
(117, 2)
```

In [11]:
```
data.columns
```

Out[11]:
```
Index(['LoanStatNew', 'Description'], dtype='object')
```

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

Out[12]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term', 'int_rate', 'installment', 'grade', 'sub_grade',
       ...
       'num_tl_90g_dpd_24m', 'num_tl_op_past_12m', 'pct_tl_nvr_dlq',
       'percent_bc_gt_75', 'pub_rec_bankruptcies', 'tax_liens',
       'tot_hi_cred_lim', 'total_bal_ex_mort', 'total_bc_limit',
       'total_il_high_credit_limit'],
      dtype='object', length=115)
```

In [13]:
```
data.dtypes
```

Out[13]:
```
LoanStatNew    object
Description    object
dtype: object
```

In [14]:
```
for i in df:
    print (i)
```

```
id
member_id
loan_amnt
funded_amnt
funded_amnt_inv
term
int_rate
installment
grade
sub_grade
emp_title
emp_length
home_ownership
annual_inc
verification_status
issue_d
loan_status
pymnt_plan
url
```

Data Cleansing

In [15]:
```
missing_data = df.isnull().mean().sort_values(ascending=False)
```

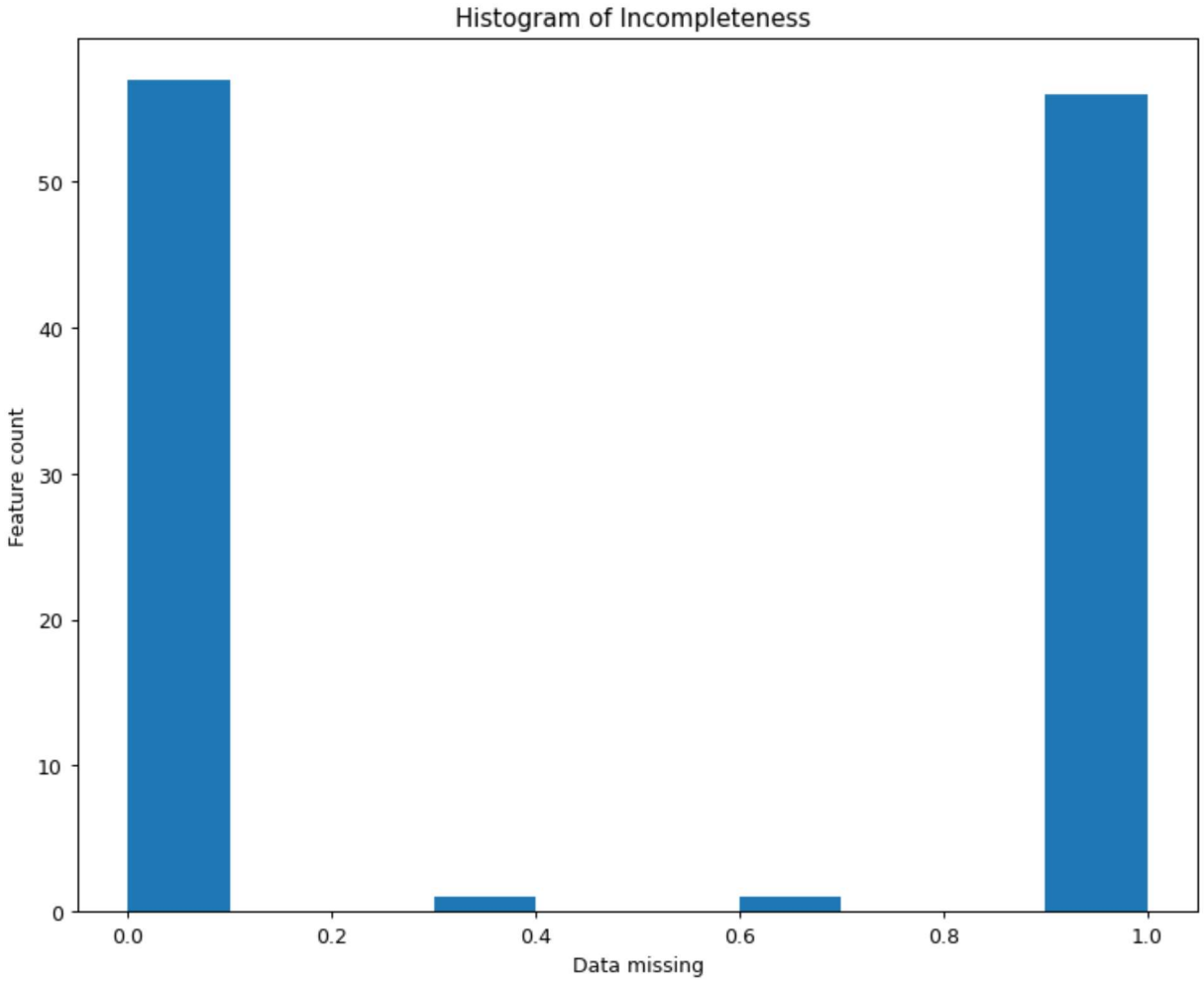
In [16]:
```
missing_data.head()
```

Out[16]:
```
annual_inc_joint          1.0
mo_sin_rcnt_rev_tl_op     1.0
mo_sin_old_il_acct        1.0
bc_util                   1.0
bc_open_to_buy            1.0
dtype: float64
```

In [17]:
```
len(missing_data)
```

Out[17]:
```
115
```

In [18]:
```
plt.figure(figsize=(10,8), dpi=90)
h= missing_data.plot.hist(bins=10)
plt.title('Histogram of Incompleteness ')
plt.xlabel('Data missing')
plt.ylabel('Feature count')
# h.FaceColor = [0.5 ,0.9];
h.EdgeColor = 'r';
```



In Above Graph show Incompleteness or Null data

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
id                                0
member_id                         3
loan_amnt                         3
funded_amnt                       3
funded_amnt_inv                   3
                                ...
tax_liens                       108
tot_hi_cred_lim               42538
total_bal_ex_mort             42538
total_bc_limit                42538
total_il_high_credit_limit    42538
Length: 115, dtype: int64
```

In [20]:
```
null_val = []
for val in df:
    if df[val].isna().sum() > 35000:
        null_val.append(val)
```

The Below data show's all Null Columns

In [21]:
```
null_val
```

Out[21]:
```
['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']
```

In [22]:
```
df[null_val]
```

Out[22]:

	mths_since_last_record	next_pymnt_d	mths_since_last_major_derog	annual_inc_joint	dti_jo
0	NaN	NaN	NaN	NaN	N
1	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	N
3	NaN	NaN	NaN	NaN	N
4	NaN	Oct-2016	NaN	NaN	N
...	...	...	...	...	
42533	NaN	Jul-2010	NaN	NaN	N
42534	NaN	Jul-2010	NaN	NaN	N
42535	NaN	Jul-2010	NaN	NaN	N
42536	NaN	NaN	NaN	NaN	N
42537	NaN	NaN	NaN	NaN	N

42538 rows × 56 columns

In [23]:
```
df.drop(columns=null_val,axis=1,inplace=True)
```

In [24]:
```
df
```

Out[24]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installı
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	16
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	5
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	8
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	33
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	6
...	...	...	...	...	...	...	...	
42533	72176	70868.0	2525.0	2525.0	225.0	36 months	9.33%	8
42534	71623	70735.0	6500.0	6500.0	0.0	36 months	8.38%	20
42535	70686	70681.0	5000.0	5000.0	0.0	36 months	7.75%	15
42536	Total amount funded in policy code 1: 460296150	NaN	NaN	NaN	NaN	NaN	NaN	
42537	Total amount funded in policy code 2: 0	NaN	NaN	NaN	NaN	NaN	NaN	

42538 rows × 59 columns

In [25]:
```
df.isnull().sum()
```

Out[25]:

id	0
member_id	3
loan_amnt	3
funded_amnt	3
funded_amnt_inv	3
term	3
int_rate	3
installment	3
grade	3
sub_grade	3
emp_title	2629
emp_length	1115
home_ownership	3
annual_inc	7
verification_status	3
issue_d	3
loan_status	3
pymnt_plan	3
url	3
desc	13296
purpose	3
title	16
zip_code	3
addr_state	3
dti	3
delinq_2yrs	32
earliest_cr_line	32
fico_range_low	3
fico_range_high	3
inq_last_6mths	32
mths_since_last_delinq	26929
open_acc	32
pub_rec	32
revol_bal	3
revol_util	93
total_acc	32
initial_list_status	3
out_prncp	3
out_prncp_inv	3
total_pymnt	3
total_pymnt_inv	3
total_rec_prncp	3
total_rec_int	3
total_rec_late_fee	3
recoveries	3
collection_recovery_fee	3
last_pymnt_d	86
last_pymnt_amnt	3
last_credit_pull_d	7
last_fico_range_high	3
last_fico_range_low	3
collections_12_mths_ex_med	148
policy_code	3
application_type	3

```
acc_now_delinq                 32
chargeoff_within_12_mths      148
delinq_amnt                    32
pub_rec_bankruptcies         1368
tax_liens                     108
dtype: int64
```

In [26]:
```
df['emp_title'].isnull().sum()
```

Out[26]:
```
2629
```

In above Data show's Total Null Columns

In [27]:
```
df['emp_title']
```

Out[27]:
```
0                            NaN
1                          Ryder
2                            NaN
3              AIR RESOURCES BOARD
4        University Medical Group
                   ...
42533                        NaN
42534                        NaN
42535                  Homemaker
42536                        NaN
42537                        NaN
Name: emp_title, Length: 42538, dtype: object
```

In [28]:
```
df['emp_title'].fillna('Other',inplace=True)
```

In [29]:
```
df['emp_title']
```

Out[29]:
```
0                          Other
1                          Ryder
2                          Other
3              AIR RESOURCES BOARD
4        University Medical Group
                   ...
42533                      Other
42534                      Other
42535                  Homemaker
42536                      Other
42537                      Other
Name: emp_title, Length: 42538, dtype: object
```

In [30]:
```
df['emp_title'].isnull().sum()
```

Out[30]:
```
0
```

In [31]:
```
df['emp_length'].isnull().sum()
```

Out[31]:
```
1115
```

In [32]:
```
df['emp_length']
```

Out[32]:
```
0        10+ years
1         < 1 year
2        10+ years
3        10+ years
4           1 year
           ...
42533     < 1 year
42534     < 1 year
42535    10+ years
42536          NaN
42537          NaN
Name: emp_length, Length: 42538, dtype: object
```

In [33]:
```
df['emp_length'].value_counts()
```

Out[33]:
```
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 [34]:
```
df['emp_length'].fillna('1 < year',inplace=True)
```

In [35]:
```
df['emp_length'].isnull().sum()
```

Out[35]:
```
0
```

In [36]:
```
df['desc']
```

Out[36]:
```
0          Borrower added on 12/22/11 > I need to upgra...
1          Borrower added on 12/22/11 > I plan to use t...
2                                                    NaN
3          Borrower added on 12/21/11 > to pay for prop...
4          Borrower added on 12/21/11 > I plan on combi...
                              ...
42533    I need to pay $2,100 for fixing my Volvo :)  A...
42534    Hi,   I'm buying  a used car. Anybody on faceb...
42535    I need to make several improvements around the...
42536                                                NaN
42537                                                NaN
Name: desc, Length: 42538, dtype: object
```

In [37]:
```
df['desc'].value_counts()
```

Out[37]:
```
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 sell o
f my home that I have up for sale and in contract. I am looking at paying $2000
0 of the difference so I can relocate closer to me and my wife's employers. We
have no other debt, to include credit cards, with the exception of an auto loan
at $19000. I plan on solely paying this debt off and ultimately purchasing a ho
me again in a closer location. I initially needed a $12000 loan , 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 aerospace
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 fen
cing, and misc other.
1
Name: desc, Length: 28963, dtype: int64
```

In [38]:
```
df['desc'].isnull().sum()
```

Out[38]:
```
13296
```

In [39]:
```
df['mths_since_last_delinq'].isnull().sum()
```

Out[39]:
```
26929
```

In [40]:
```
df['mths_since_last_delinq']
```

Out[40]:
```
0         NaN
1         NaN
2         NaN
3        35.0
4        38.0
         ...
42533     NaN
42534     NaN
42535     NaN
42536     NaN
42537     NaN
Name: mths_since_last_delinq, Length: 42538, dtype: float64
```

In [41]:
```
df['url'].isnull().sum()
```

Out[41]:
```
3
```

In [42]:
```
df.drop(columns=['mths_since_last_delinq','desc','url'],inplace=True)
```

In [43]:
```
df.isnull().sum()
```

Out[43]:

id	0
member_id	3
loan_amnt	3
funded_amnt	3
funded_amnt_inv	3
term	3
int_rate	3
installment	3
grade	3
sub_grade	3
emp_title	0
emp_length	0
home_ownership	3
annual_inc	7
verification_status	3
issue_d	3
loan_status	3
pymnt_plan	3
purpose	3
title	16
zip_code	3
addr_state	3
dti	3
delinq_2yrs	32
earliest_cr_line	32
fico_range_low	3
fico_range_high	3
inq_last_6mths	32
open_acc	32
pub_rec	32
revol_bal	3
revol_util	93
total_acc	32
initial_list_status	3
out_prncp	3
out_prncp_inv	3
total_pymnt	3
total_pymnt_inv	3
total_rec_prncp	3
total_rec_int	3
total_rec_late_fee	3
recoveries	3
collection_recovery_fee	3
last_pymnt_d	86
last_pymnt_amnt	3
last_credit_pull_d	7
last_fico_range_high	3
last_fico_range_low	3
collections_12_mths_ex_med	148
policy_code	3
application_type	3
acc_now_delinq	32
chargeoff_within_12_mths	148
delinq_amnt	32

```
pub_rec_bankruptcies          1368
tax_liens                      108
dtype: int64
```

In [44]:
```
df['pub_rec_bankruptcies'].isnull().sum()
```

Out[44]:
```
1368
```

In [45]:
```
df['pub_rec_bankruptcies'].value_counts()
```

Out[45]:
```
0.0    39316
1.0     1846
2.0        8
Name: pub_rec_bankruptcies, dtype: int64
```

In [46]:
```
df['pub_rec_bankruptcies'].fillna(0,inplace=True)
```

In [47]:
```
df.isnull().sum()
```

Out[47]:

id	0
member_id	3
loan_amnt	3
funded_amnt	3
funded_amnt_inv	3
term	3
int_rate	3
installment	3
grade	3
sub_grade	3
emp_title	0
emp_length	0
home_ownership	3
annual_inc	7
verification_status	3
issue_d	3
loan_status	3
pymnt_plan	3
purpose	3
title	16
zip_code	3
addr_state	3
dti	3
delinq_2yrs	32
earliest_cr_line	32
fico_range_low	3
fico_range_high	3
inq_last_6mths	32
open_acc	32
pub_rec	32
revol_bal	3
revol_util	93
total_acc	32
initial_list_status	3
out_prncp	3
out_prncp_inv	3
total_pymnt	3
total_pymnt_inv	3
total_rec_prncp	3
total_rec_int	3
total_rec_late_fee	3
recoveries	3
collection_recovery_fee	3
last_pymnt_d	86
last_pymnt_amnt	3
last_credit_pull_d	7
last_fico_range_high	3
last_fico_range_low	3
collections_12_mths_ex_med	148
policy_code	3
application_type	3
acc_now_delinq	32
chargeoff_within_12_mths	148
delinq_amnt	32

```
pub_rec_bankruptcies           0
tax_liens                    108
dtype: int64
```

In [48]:
```
df['collections_12_mths_ex_med'].isnull().sum()
```

Out[48]:
```
148
```

In [49]:
```
df['chargeoff_within_12_mths'].isnull().sum()
```

Out[49]:
```
148
```

In [50]:
```
df['collections_12_mths_ex_med'].value_counts()
```

Out[50]:
```
0.0    42390
Name: collections_12_mths_ex_med, dtype: int64
```

In [51]:
```
df['chargeoff_within_12_mths'].value_counts()
```

Out[51]:
```
0.0    42390
Name: chargeoff_within_12_mths, dtype: int64
```

In [52]:
```
df['tax_liens'].isnull().sum()
```

Out[52]:
```
108
```

In [53]:
```
df['tax_liens'].value_counts()
```

Out[53]:
```
0.0    42429
1.0        1
Name: tax_liens, dtype: int64
```

In [54]:
```
df.drop(columns=['collections_12_mths_ex_med','chargeoff_within_12_mths','tax_lie
```

In [55]:
```
df.isnull().sum()
```

Out[55]:
```
id                          0
member_id                   3
loan_amnt                   3
funded_amnt                 3
funded_amnt_inv             3
term                        3
int_rate                    3
installment                 3
grade                       3
sub_grade                   3
emp_title                   0
emp_length                  0
home_ownership              3
annual_inc                  7
verification_status         3
issue_d                     3
loan_status                 3
pymnt_plan                  3
purpose                     3
title                      16
zip_code                    3
addr_state                  3
dti                         3
delinq_2yrs                32
earliest_cr_line           32
fico_range_low              3
fico_range_high             3
inq_last_6mths             32
open_acc                   32
pub_rec                    32
revol_bal                   3
revol_util                 93
total_acc                  32
initial_list_status         3
out_prncp                   3
out_prncp_inv               3
total_pymnt                 3
total_pymnt_inv             3
total_rec_prncp             3
total_rec_int               3
total_rec_late_fee          3
recoveries                  3
collection_recovery_fee     3
last_pymnt_d               86
last_pymnt_amnt             3
last_credit_pull_d          7
last_fico_range_high        3
last_fico_range_low         3
policy_code                 3
application_type            3
acc_now_delinq             32
delinq_amnt                32
pub_rec_bankruptcies        0
dtype: int64
```

In [56]:
```
df['last_credit_pull_d'].value_counts()
```

Out[56]:
```
Sep-2016    16232
Mar-2016      859
Aug-2016      771
Apr-2016      700
Feb-2013      696
              ...
Nov-2007        3
May-2008        1
Jul-2008        1
Jun-2008        1
Jul-2007        1
Name: last_credit_pull_d, Length: 111, dtype: int64
```

In [57]:
```
df['last_pymnt_d'].value_counts()
```

Out[57]:
```
Mar-2013    1070
Dec-2014     949
May-2013     943
Feb-2013     906
Mar-2012     893
             ...
Jun-2008      20
Mar-2008      18
Jan-2008      11
Feb-2008       8
Dec-2007       2
Name: last_pymnt_d, Length: 106, dtype: int64
```

In [58]:
```
df['last_pymnt_d'].isnull().sum()
```

Out[58]:
```
86
```

In [59]:
```
df['last_pymnt_amnt'].value_counts()
```

Out[59]:
```
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 [60]:
```
df['last_pymnt_amnt'].isnull().sum()
```

Out[60]:
```
3
```

In [61]:
```
df.drop(columns=['last_pymnt_amnt','last_pymnt_d'],axis=1,inplace=True)
```

In [62]:
```
df.isnull().sum()
```

Out[62]:
```
id                          0
member_id                   3
loan_amnt                   3
funded_amnt                 3
funded_amnt_inv             3
term                        3
int_rate                    3
installment                 3
grade                       3
sub_grade                   3
emp_title                   0
emp_length                  0
home_ownership              3
annual_inc                  7
verification_status         3
issue_d                     3
loan_status                 3
pymnt_plan                  3
purpose                     3
title                      16
zip_code                    3
addr_state                  3
dti                         3
delinq_2yrs                32
earliest_cr_line           32
fico_range_low              3
fico_range_high             3
inq_last_6mths             32
open_acc                   32
pub_rec                    32
revol_bal                   3
revol_util                 93
total_acc                  32
initial_list_status         3
out_prncp                   3
out_prncp_inv               3
total_pymnt                 3
total_pymnt_inv             3
total_rec_prncp             3
total_rec_int               3
total_rec_late_fee          3
recoveries                  3
collection_recovery_fee     3
last_credit_pull_d          7
last_fico_range_high        3
last_fico_range_low         3
policy_code                 3
application_type            3
acc_now_delinq             32
delinq_amnt                32
pub_rec_bankruptcies        0
dtype: int64
```

In [63]:
```
df['title'].isnull().sum()
```

Out[63]:
```
16
```

In [64]:
```
df['title'].value_counts()
```

Out[64]:
```
Debt Consolidation                  2259
Debt Consolidation Loan             1760
Personal Loan                        708
Consolidation                        547
debt consolidation                   532
                                    ...
CitiCard PayOff                        1
Taxes Loan                             1
Blazing in 5 years                     1
I was scammed and now recovering       1
Aroundthehouse                         1
Name: title, Length: 21264, dtype: int64
```

In [65]:
```
df['title'].fillna('Other',inplace=True)
```

In [66]:
```
df['title'].isnull().sum()
```

Out[66]:
```
0
```

In [67]:
```
df['delinq_2yrs'].value_counts()
```

Out[67]:
```
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 [68]:
```
df['delinq_2yrs'].isnull().sum()
```

Out[68]:
```
32
```

In [69]:
```
df['delinq_2yrs'].isna().sum()
```

Out[69]:
```
32
```

In [70]:
```
df['delinq_2yrs'].fillna(0,inplace=True)
```

In [71]:
```
df['delinq_2yrs'].isnull().sum()
```

Out[71]:
```
0
```

In [72]:
```
df['earliest_cr_line'].isnull().sum()
```

Out[72]:
```
32
```

In [73]:
```
df['earliest_cr_line'].fillna(0,inplace=True)
```

In [74]:
```
df['earliest_cr_line'].isnull().sum()
```

Out[74]:
```
0
```

In [75]:
```
df['earliest_cr_line'].value_counts()
```

Out[75]:
```
Oct-1999    393
Nov-1998    390
Oct-2000    370
Dec-1998    366
Dec-1997    348
            ...
Mar-1963      1
Nov-1954      1
Jun-1959      1
Dec-1965      1
Nov-1959      1
Name: earliest_cr_line, Length: 531, dtype: int64
```

In [76]:
```
df['revol_util'].value_counts()
```

Out[76]:
```
0%        1070
40.7%       65
0.2%        64
63%         63
66.6%       62
            ...
21.59%       1
39.95%       1
0.01%        1
17.67%       1
105.7%       1
Name: revol_util, Length: 1119, dtype: int64
```

In [77]:
```
df['revol_bal'].value_counts()
```

Out[77]:
```
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 [78]:
```
df['revol_util'].isnull().sum()
```

Out[78]:
```
93
```

In [79]:
```
df['revol_bal'].isnull().sum()
```

Out[79]:
```
3
```

In [80]:
```
df['revol_util'].fillna('0%',inplace=True)
```

In [81]:
```
df['revol_util'].isnull().sum()
```

Out[81]:
```
0
```

In [82]:
```
df['revol_bal'].fillna('0.0',inplace=True)
```

In [83]:
```
df['revol_bal'].isnull().sum()
```

Out[83]:
```
0
```

In [84]:
```
df.isnull().sum()
```

Out[84]:
```
id                          0
member_id                   3
loan_amnt                   3
funded_amnt                 3
funded_amnt_inv             3
term                        3
int_rate                    3
installment                 3
grade                       3
sub_grade                   3
emp_title                   0
emp_length                  0
home_ownership              3
annual_inc                  7
verification_status         3
issue_d                     3
loan_status                 3
pymnt_plan                  3
purpose                     3
title                       0
zip_code                    3
addr_state                  3
dti                         3
delinq_2yrs                 0
earliest_cr_line            0
fico_range_low              3
fico_range_high             3
inq_last_6mths             32
open_acc                   32
pub_rec                    32
revol_bal                   0
revol_util                  0
total_acc                  32
initial_list_status         3
out_prncp                   3
out_prncp_inv               3
total_pymnt                 3
total_pymnt_inv             3
total_rec_prncp             3
total_rec_int               3
total_rec_late_fee          3
recoveries                  3
collection_recovery_fee     3
last_credit_pull_d          7
last_fico_range_high        3
last_fico_range_low         3
policy_code                 3
application_type            3
acc_now_delinq             32
delinq_amnt                32
pub_rec_bankruptcies        0
dtype: int64
```

In [85]:
```
df['member_id'].value_counts()
```

Out[85]:
```
1296599.0    1
694920.0     1
697615.0     1
697589.0     1
697502.0     1
             ..
971607.0     1
970659.0     1
971558.0     1
966956.0     1
70681.0      1
Name: member_id, Length: 42535, dtype: int64
```

In [86]:
```
df['member_id'].fillna(0,inplace=True)
```

In [87]:
```
df['loan_amnt'].value_counts()
```

Out[87]:
```
10000.0    3016
12000.0    2439
5000.0     2260
6000.0     2037
15000.0    2012
           ...
10350.0       1
19100.0       1
17975.0       1
31150.0       1
20425.0       1
Name: loan_amnt, Length: 898, dtype: int64
```

In [88]:
```
df['loan_status'].value_counts()
```

Out[88]:
```
Fully Paid                                            33586
Charged Off                                            5653
Does not meet the credit policy. Status:Fully Paid     1988
Does not meet the credit policy. Status:Charged Off     761
Current                                                 513
In Grace Period                                          16
Late (31-120 days)                                       12
Late (16-30 days)                                         5
Default                                                   1
Name: loan_status, dtype: int64
```

In [89]:
```
df['loan_amnt'].fillna('0',inplace=True)
```

In [90]:
```
df['loan_amnt'].isnull().sum()
```

Out[90]:
```
0
```

In [91]:
```
df.isnull().sum()
```

Out[91]:
```
id                          0
member_id                   0
loan_amnt                   0
funded_amnt                 3
funded_amnt_inv             3
term                        3
int_rate                    3
installment                 3
grade                       3
sub_grade                   3
emp_title                   0
emp_length                  0
home_ownership              3
annual_inc                  7
verification_status         3
issue_d                     3
loan_status                 3
pymnt_plan                  3
purpose                     3
title                       0
zip_code                    3
addr_state                  3
dti                         3
delinq_2yrs                 0
earliest_cr_line            0
fico_range_low              3
fico_range_high             3
inq_last_6mths             32
open_acc                   32
pub_rec                    32
revol_bal                   0
revol_util                  0
total_acc                  32
initial_list_status         3
out_prncp                   3
out_prncp_inv               3
total_pymnt                 3
total_pymnt_inv             3
total_rec_prncp             3
total_rec_int               3
total_rec_late_fee          3
recoveries                  3
collection_recovery_fee     3
last_credit_pull_d          7
last_fico_range_high        3
last_fico_range_low         3
policy_code                 3
application_type            3
acc_now_delinq             32
delinq_amnt                32
pub_rec_bankruptcies        0
dtype: int64
```

In [92]:
```
df.dropna(axis=0, inplace=True)
```

In [93]:
```
df.head()
```

Out[93]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment
0	1077501	1296599.0	5000.0	5000.0	4975.0	36 months	10.65%	162.87
1	1077430	1314167.0	2500.0	2500.0	2500.0	60 months	15.27%	59.83
2	1077175	1313524.0	2400.0	2400.0	2400.0	36 months	15.96%	84.33
3	1076863	1277178.0	10000.0	10000.0	10000.0	36 months	13.49%	339.31
4	1075358	1311748.0	3000.0	3000.0	3000.0	60 months	12.69%	67.79

5 rows × 51 columns

In [94]:
```
df['member_id'] = df['member_id'].astype('int')
```

In [95]:
```
df['term'].value_counts()
```

Out[95]:
```
 36 months    31502
 60 months    11001
Name: term, dtype: int64
```

In [96]:
```
df['term'].replace({' 36 months':36, ' 60 months':60},inplace=True)
```

In [97]:
```
df.rename(columns={'term':'term_of_months'},inplace=True)
```

In [98]:
```
df['int_rate'] = list(map(lambda x: x[:-1], df['int_rate'].values))
```

In [99]:
```
df.columns
```

Out[99]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term_of_months', 'int_rate', 'installment', 'grade', 'sub_grade',
       'emp_title', 'emp_length', 'home_ownership', 'annual_inc',
       'verification_status', 'issue_d', 'loan_status', 'pymnt_plan',
       'purpose', 'title', 'zip_code', 'addr_state', 'dti', 'delinq_2yrs',
       'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'initial_list_status', '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_credit_pull_d', 'last_fico_range_high', 'last_fico_range_low',
       'policy_code', 'application_type', 'acc_now_delinq', 'delinq_amnt',
       'pub_rec_bankruptcies'],
      dtype='object')
```

In [100]:
```
df.rename(columns={'int_rate':'int_rate%'},inplace=True)
```

In [101]:
```
df.head()
```

Out[101]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term_of_months	int_rate%	in
0	1077501	1296599	5000.0	5000.0	4975.0	36	10.65	
1	1077430	1314167	2500.0	2500.0	2500.0	60	15.27	
2	1077175	1313524	2400.0	2400.0	2400.0	36	15.96	
3	1076863	1277178	10000.0	10000.0	10000.0	36	13.49	
4	1075358	1311748	3000.0	3000.0	3000.0	60	12.69	

5 rows × 51 columns

In [102]:
```
df.isnull().sum()
```

Out[102]:
```
id                          0
member_id                   0
loan_amnt                   0
funded_amnt                 0
funded_amnt_inv             0
term_of_months              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
pymnt_plan                  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
initial_list_status         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_credit_pull_d          0
last_fico_range_high        0
last_fico_range_low         0
policy_code                 0
application_type            0
acc_now_delinq              0
delinq_amnt                 0
pub_rec_bankruptcies        0
dtype: int64
```

In Above data there is no Null Data

In [103]:
```
df[['emp_length']]
```

Out[103]:

	emp_length
0	10+ years
1	< 1 year
2	10+ years
3	10+ years
4	1 year
...	...
42509	1 year
42511	1 year
42512	7 years
42513	< 1 year
42514	2 years

42503 rows × 1 columns

In [104]:
```
df['emp_length'] = df['emp_length'].replace({'< 1 year':'0'})
```

In [105]:
```
df.rename(columns={'emp_length':'emp_length_years'},inplace=True)
```

In [106]:
```
df[['emp_length_years']]
```

Out[106]:

	emp_length_years
0	10+ years
1	0
2	10+ years
3	10+ years
4	1 year
...	...
42509	1 year
42511	1 year
42512	7 years
42513	0
42514	2 years

42503 rows × 1 columns

In [107]:
```
df['emp_length_years']=df['emp_length_years'].str.replace('years',"",regex=True)
```

In [108]:
```
df['emp_length_years']=df['emp_length_years'].str.replace('year',"",regex=True)
```

In [109]:
```
df['emp_length_years']=df['emp_length_years'].str.replace('+',"",regex=True)
```

In [110]:
```
df[['emp_length_years']]
```

Out[110]:

	emp_length_years
0	10
1	0
2	10
3	10
4	1
...	...
42509	1
42511	1
42512	7
42513	0
42514	2

42503 rows × 1 columns

In [ ]:
```

```

In [111]:
```
df['revol_util'] = list(map(lambda x: x[:-1], df['revol_util'].values))
```

In [112]:
```
df[['revol_util']]
```

Out[112]:

	revol_util
0	83.7
1	9.4
2	98.5
3	21
4	53.9
...	...
42509	0
42511	85
42512	2.2
42513	66
42514	63.5

42503 rows × 1 columns

In [113]:
```
df['total_pymnt']=df['total_pymnt'].round(2)
```

In [114]:
```
df['total_pymnt'].dtypes
```

Out[114]:
```
dtype('float64')
```

In [115]:
```
df[['total_pymnt']]
```

Out[115]:

	total_pymnt
0	5863.16
1	1008.71
2	3005.67
3	12231.89
4	3784.49
...	...
42509	6486.77
42511	12622.32
42512	2227.02
42513	7029.87
42514	5084.72

42503 rows × 1 columns

In [116]:
```
df.shape
```

Out[116]:
```
(42503, 51)
```

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

Out[117]:
```
id                          0
member_id                   0
loan_amnt                   0
funded_amnt                 0
funded_amnt_inv             0
term_of_months              0
int_rate%                   0
installment                 0
grade                       0
sub_grade                   0
emp_title                   0
emp_length_years            0
home_ownership              0
annual_inc                  0
verification_status         0
issue_d                     0
loan_status                 0
pymnt_plan                  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
initial_list_status         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_credit_pull_d          0
last_fico_range_high        0
last_fico_range_low         0
policy_code                 0
application_type            0
acc_now_delinq              0
delinq_amnt                 0
pub_rec_bankruptcies        0
dtype: int64
```

In [118]:
```
df.head()
```

Out[118]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term_of_months	int_rate%	in
0	1077501	1296599	5000.0	5000.0	4975.0	36	10.65	
1	1077430	1314167	2500.0	2500.0	2500.0	60	15.27	
2	1077175	1313524	2400.0	2400.0	2400.0	36	15.96	
3	1076863	1277178	10000.0	10000.0	10000.0	36	13.49	
4	1075358	1311748	3000.0	3000.0	3000.0	60	12.69	

5 rows × 51 columns

In [119]:
```
pd.crosstab(df.verification_status,df.home_ownership)
```

Out[119]:

home_ownership	MORTGAGE	NONE	OTHER	OWN	RENT
verification_status					
Not Verified	7792	4	74	1620	9236
Source Verified	4155	0	14	792	5345
Verified	7004	0	48	838	5581

In [ ]:
```

```

In [120]:
```
pd.crosstab(df.verification_status,df.pymnt_plan)
```

Out[120]:

pymnt_plan	n	y
verification_status		
Not Verified	18726	0
Source Verified	10306	0
Verified	13470	1

In [ ]:
```

```

In [121]:
```
pd.crosstab(df.purpose,df.pymnt_plan)
```

Out[121]:

pymnt_plan	n	y
purpose		
car	1615	0
credit_card	5477	0
debt_consolidation	19773	1
educational	422	0
home_improvement	3199	0
house	426	0
major_purchase	2311	0
medical	753	0
moving	629	0
other	4396	0
renewable_energy	106	0
small_business	1991	0
vacation	400	0
wedding	1004	0

In [ ]:
```

```

In [122]:
```
pd.crosstab(df.pymnt_plan,df.emp_length_years)
```

Out[122]:

emp_length_years	0	1	1 <	10	2	3	4	5	6	7	8	9
pymnt_plan												
n	5043	3590	1112	9366	4742	4362	3649	3457	2374	1875	1592	1340
y	0	1	0	0	0	0	0	0	0	0	0	0

In [ ]:
```

```

In [123]:
```
df.columns
```

Out[123]:
```
Index(['id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv',
       'term_of_months', 'int_rate%', 'installment', 'grade', 'sub_grade',
       'emp_title', 'emp_length_years', 'home_ownership', 'annual_inc',
       'verification_status', 'issue_d', 'loan_status', 'pymnt_plan',
       'purpose', 'title', 'zip_code', 'addr_state', 'dti', 'delinq_2yrs',
       'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'initial_list_status', '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_credit_pull_d', 'last_fico_range_high', 'last_fico_range_low',
       'policy_code', 'application_type', 'acc_now_delinq', 'delinq_amnt',
       'pub_rec_bankruptcies'],
      dtype='object')
```

In Below Data Not required data is stored in a variable

In [124]:
```
to_remove = ['id','member_id', 'funded_amnt', 'funded_amnt_inv','sub_grade','emp_
               'issue_d','purpose','title','zip_code','addr_state','zip_code','ou
               'total_pymnt_inv','total_rec_prncp','total_rec_int','total_rec_lat
```

In [125]:
```
to_remove
```

Out[125]:
```
['id',
 'member_id',
 'funded_amnt',
 'funded_amnt_inv',
 'sub_grade',
 'emp_title',
 'issue_d',
 'purpose',
 'title',
 'zip_code',
 'addr_state',
 'zip_code',
 'out_prncp',
 'out_prncp_inv',
 'total_pymnt',
 'total_pymnt_inv',
 'total_rec_prncp',
 'total_rec_int',
 'total_rec_late_fee',
 'recoveries',
 'collection_recovery_fee']
```

In [126]:
```
df.drop(columns=to_remove,axis=1,inplace=True)
```

In [127]:
```
df
```

Out[127]:

	loan_amnt	term_of_months	int_rate%	installment	grade	emp_length_years	home_owners
0	5000.0	36	10.65	162.87	B	10	RE
1	2500.0	60	15.27	59.83	C	0	RE
2	2400.0	36	15.96	84.33	C	10	RE
3	10000.0	36	13.49	339.31	C	10	RE
4	3000.0	60	12.69	67.79	B	1	RE
...	...	...	...	...	...	...	
42509	5350.0	36	13.12	180.57	D	1	O\
42511	10000.0	36	14.70	345.18	E	1	RE
42512	2000.0	36	7.12	61.87	A	7	MORTGA
42513	6000.0	36	10.59	195.28	C	0	RE
42514	4400.0	36	9.64	141.25	B	2	MORTGA

42503 rows × 31 columns

In [128]:
```
df.columns
```

Out[128]:
```
Index(['loan_amnt', 'term_of_months', 'int_rate%', 'installment', 'grade',
       'emp_length_years', 'home_ownership', 'annual_inc',
       'verification_status', 'loan_status', 'pymnt_plan', 'dti',
       'delinq_2yrs', 'earliest_cr_line', 'fico_range_low', 'fico_range_high',
       'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
       'total_acc', 'initial_list_status', 'last_credit_pull_d',
       'last_fico_range_high', 'last_fico_range_low', 'policy_code',
       'application_type', 'acc_now_delinq', 'delinq_amnt',
       'pub_rec_bankruptcies'],
      dtype='object')
```

In [129]:
```
data
```

Out[129]:

	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...
...	...	...
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...

117 rows × 2 columns

In [130]:
```
data['LoanStatNew'].value_counts()
```

Out[130]:
```
acc_now_delinq          1
pub_rec_bankruptcies    1
policy_code             1
percent_bc_gt_75        1
pct_tl_nvr_dlq          1
                       ..
int_rate                1
installment             1
inq_last_6mths          1
inq_last_12m            1
zip_code                1
Name: LoanStatNew, Length: 115, dtype: int64
```

In [131]:
```
data['LoanStatNew'].isnull().sum()
```

Out[131]:
```
2
```

In [132]:
```
data['Description'].value_counts()
```

Out[132]:
```
The number of accounts on which the borrower is now delinquent.
1
The number of open credit lines in the borrower's credit file.
1
Number of derogatory public records
1
publicly available policy_code=1\nnew products not publicly available policy_co
de=2    1
Percentage of all bankcard accounts > 75% of limit.
1

..
Interest Rate on the loan
1
The monthly payment owed by the borrower if the loan originates.
1
The number of inquiries in past 6 months (excluding auto and mortgage inquirie
s)       1
Number of credit inquiries in past 12 months
1
* Employer Title replaces Employer Name for all loans listed after 9/23/2013
1
Name: Description, Length: 116, dtype: int64
```

In [133]:
```
data['Description'].isnull().sum()
```

Out[133]:
```
1
```

In [134]:
```
data['LoanStatNew'].fillna('Other',inplace=True)
```

In [135]:
```
data['LoanStatNew'].isnull().sum()
```

Out[135]:
```
0
```

In [136]:
```
data['Description'].fillna("None",inplace=True)
```

In [137]:
```
data['Description'].isnull().sum()
```

Out[137]:
```
0
```

In [138]:
```
data
```

Out[138]:

	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...
...	...	...
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	Other	None
116	Other	* Employer Title replaces Employer Name for al...

117 rows × 2 columns

In [139]:
```
df.sample(8)
```

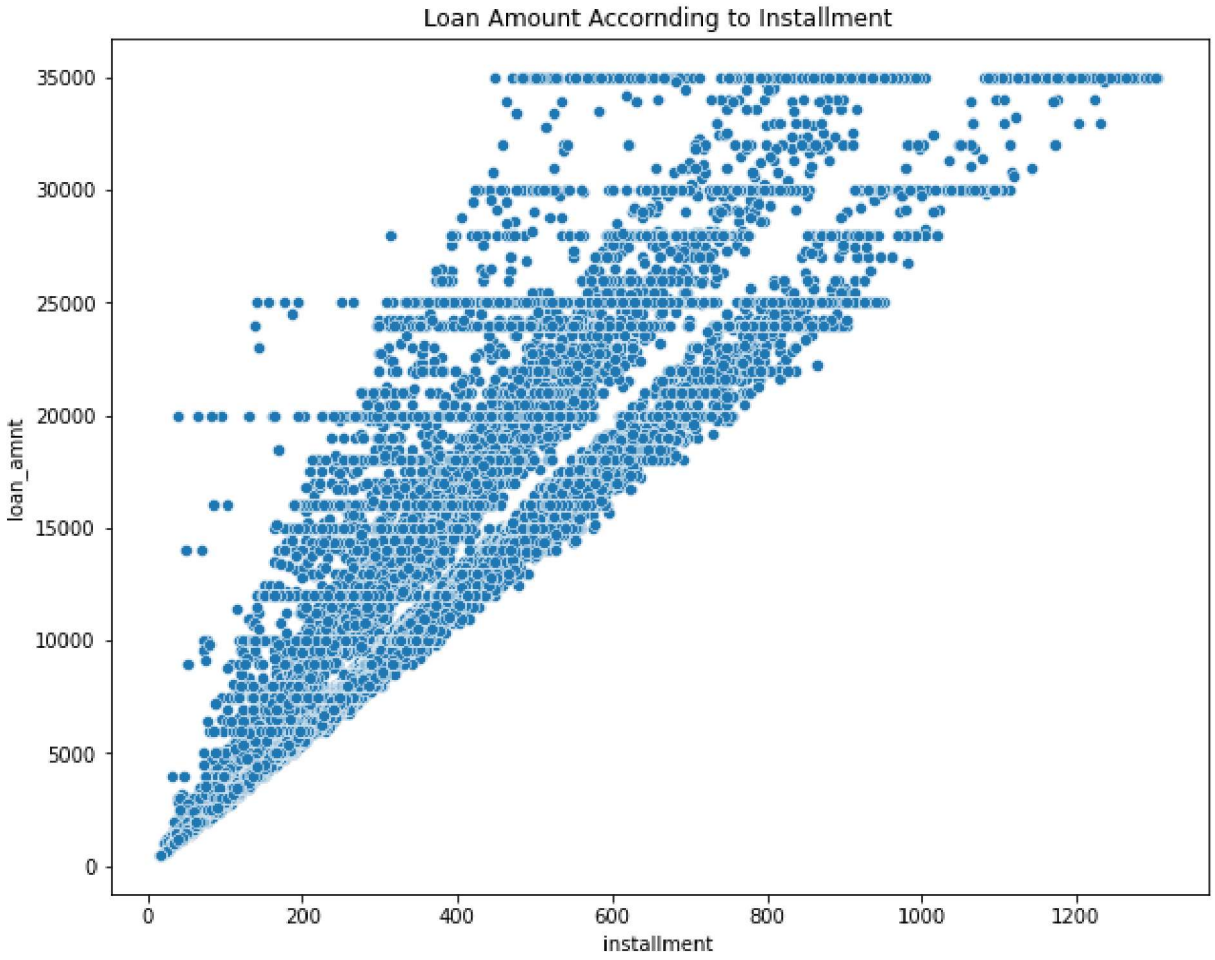
Out[139]:

	loan_amnt	term_of_months	int_rate%	installment	grade	emp_length_years	home_owners
41746	9000.0	36	8.63	284.66	A	1	RE
42202	15000.0	36	15.65	524.77	F	3	MORTGA
774	14000.0	36	8.90	444.55	A	5	RE
4881	4000.0	36	6.62	122.82	A	1 <	MORTGA
21656	18000.0	60	17.06	447.93	E	9	O\
34870	8000.0	36	16.00	281.26	D	8	OTH
41458	6500.0	36	15.70	227.56	E	1	MORTGA
16967	13000.0	36	7.29	403.13	A	6	RE

8 rows × 31 columns

Data Visualization

In [140]:
```
fig = plt.figure(figsize=(10,8))
sns.scatterplot(data=df,x='installment',y='loan_amnt',palette='tab10').set(title=
plt.show()
```



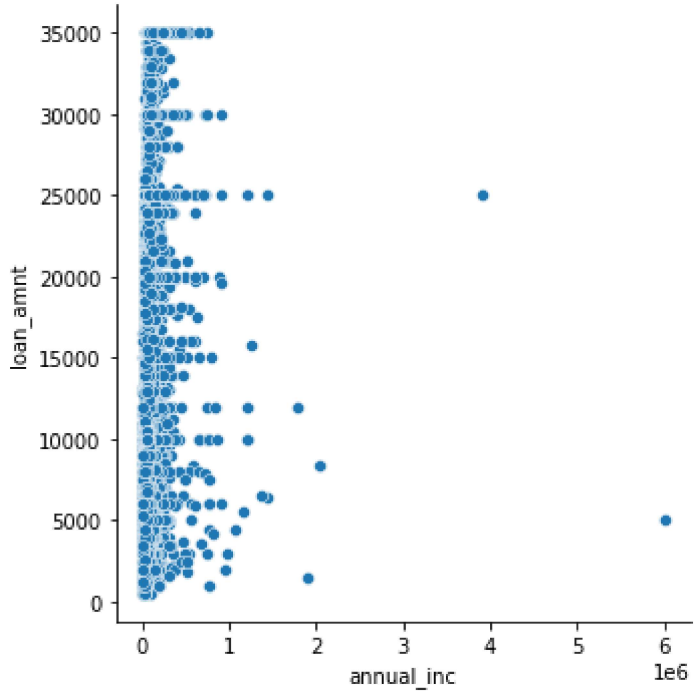
In Above Graph we can see that relation between Installment and Loan_Amount

-- In above scatter plot Minimum Loan amount 5,000 have low Installment i.e 200

-- In above scatter plot Maximum Loan amount 35,000 have High Installment i.e 1,200

In [141]:
```
sns.relplot(x='annual_inc',y='loan_amnt',data=df)
```

Out[141]:
```
<seaborn.axisgrid.FacetGrid at 0x20450dbcb80>
```

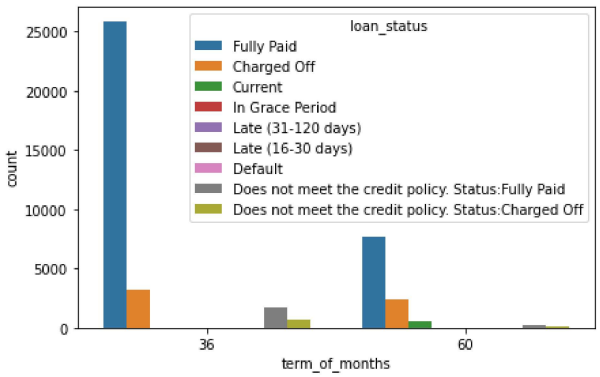


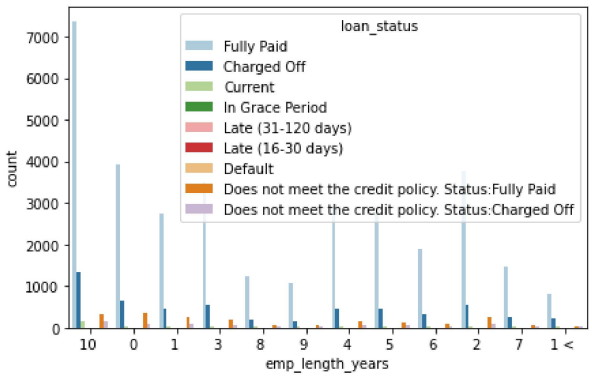
In [ ]:
```
```

In [142]:
```
plt.figure(figsize=(15, 20))

plt.subplot(4, 2, 1)
sns.countplot(x='term_of_months', data=df, hue='loan_status',palette='tab10')

plt.subplot(4, 2, 2)
sns.countplot(x='emp_length_years', data=df, hue='loan_status',palette='Paired')
plt.show()
```

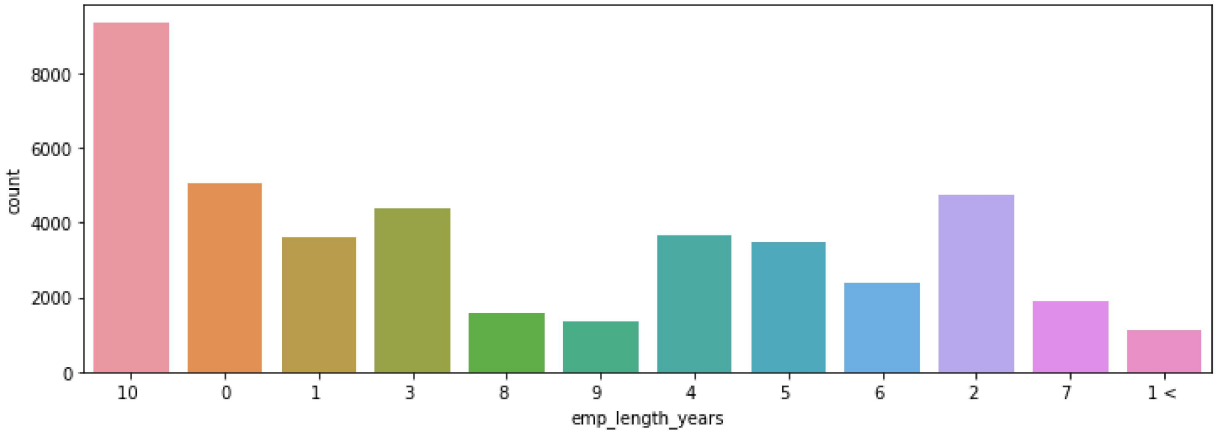




In [ ]:

In [143]:
```
plt.figure(figsize=(12,4))
sns.countplot(x='emp_length_years',data=df )
```

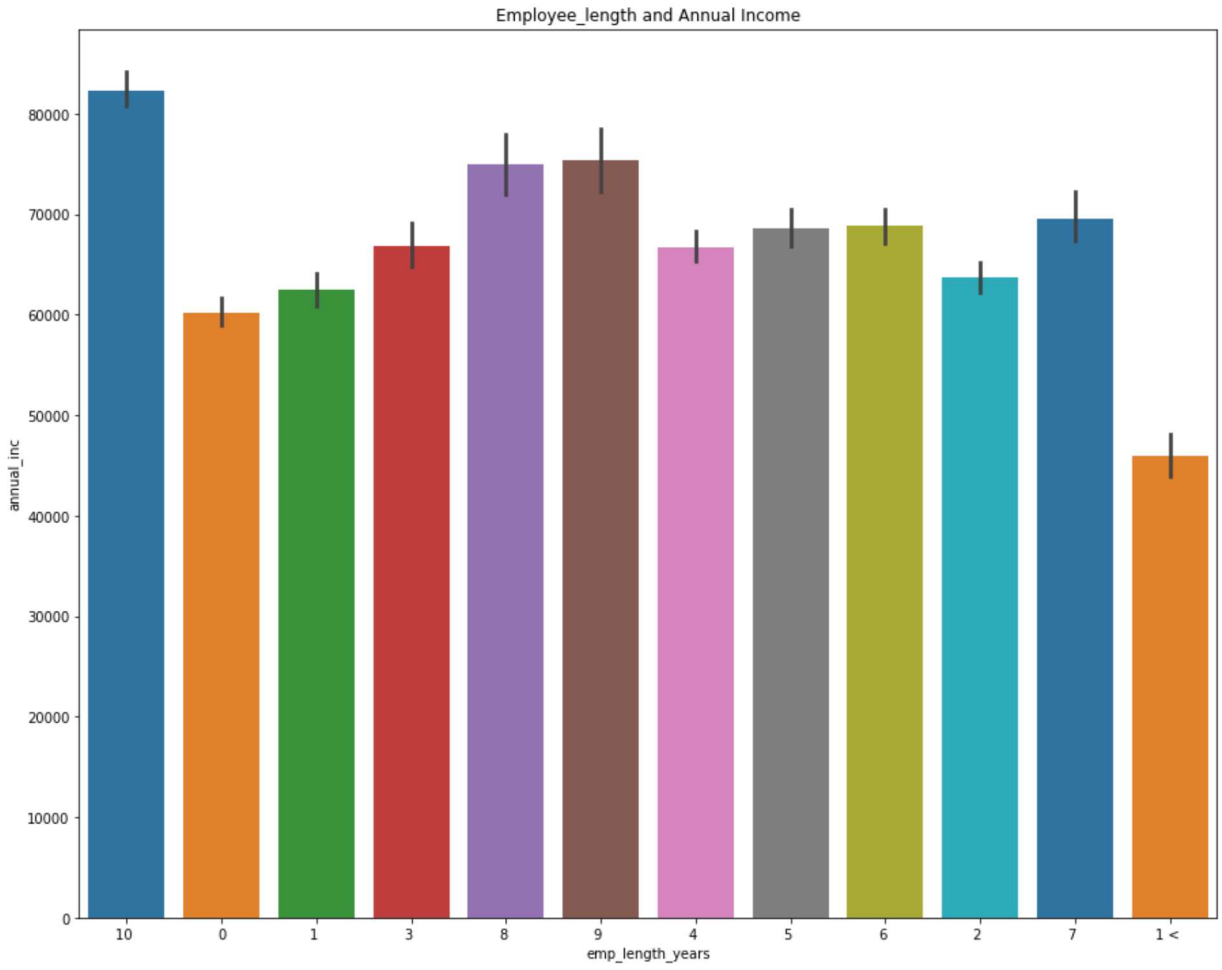
Out[143]: <AxesSubplot:xlabel='emp_length_years', ylabel='count'>



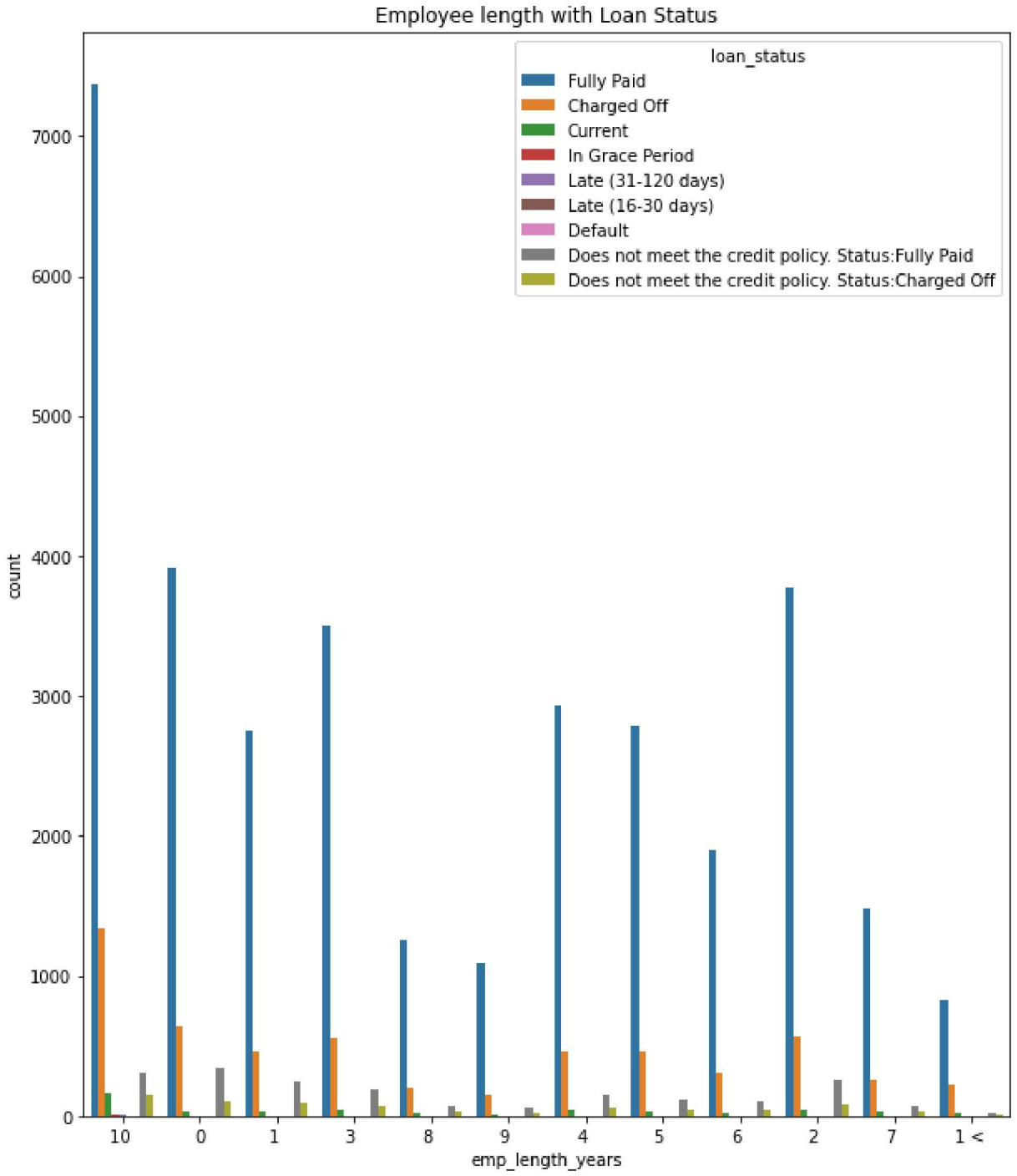
In [ ]:

In [144]:
```
fig = plt.figure(figsize=(15,12))
sns.barplot(data=df,x='emp_length_years',y='annual_inc',palette='tab10',dodge=Tru
```

Out[144]:
```
[Text(0.5, 1.0, 'Employee_length and Annual Income')]
```



In [145]:
```
plt.figure(figsize=(10,12))
sns.countplot(x='emp_length_years',data=df,hue='loan_status',linewidth=5.5).set(t
plt.show()
```



In [146]:
```
df['loan_status'].value_counts()
```

Out[146]:
```
Fully Paid                                            33585
Charged Off                                            5652
Does not meet the credit policy. Status:Fully Paid     1961
Does not meet the credit policy. Status:Charged Off     758
Current                                                 513
In Grace Period                                          16
Late (31-120 days)                                       12
Late (16-30 days)                                         5
Default                                                   1
Name: loan_status, dtype: int64
```

In [147]:
```
df.reset_index(drop=True,inplace=True)
```

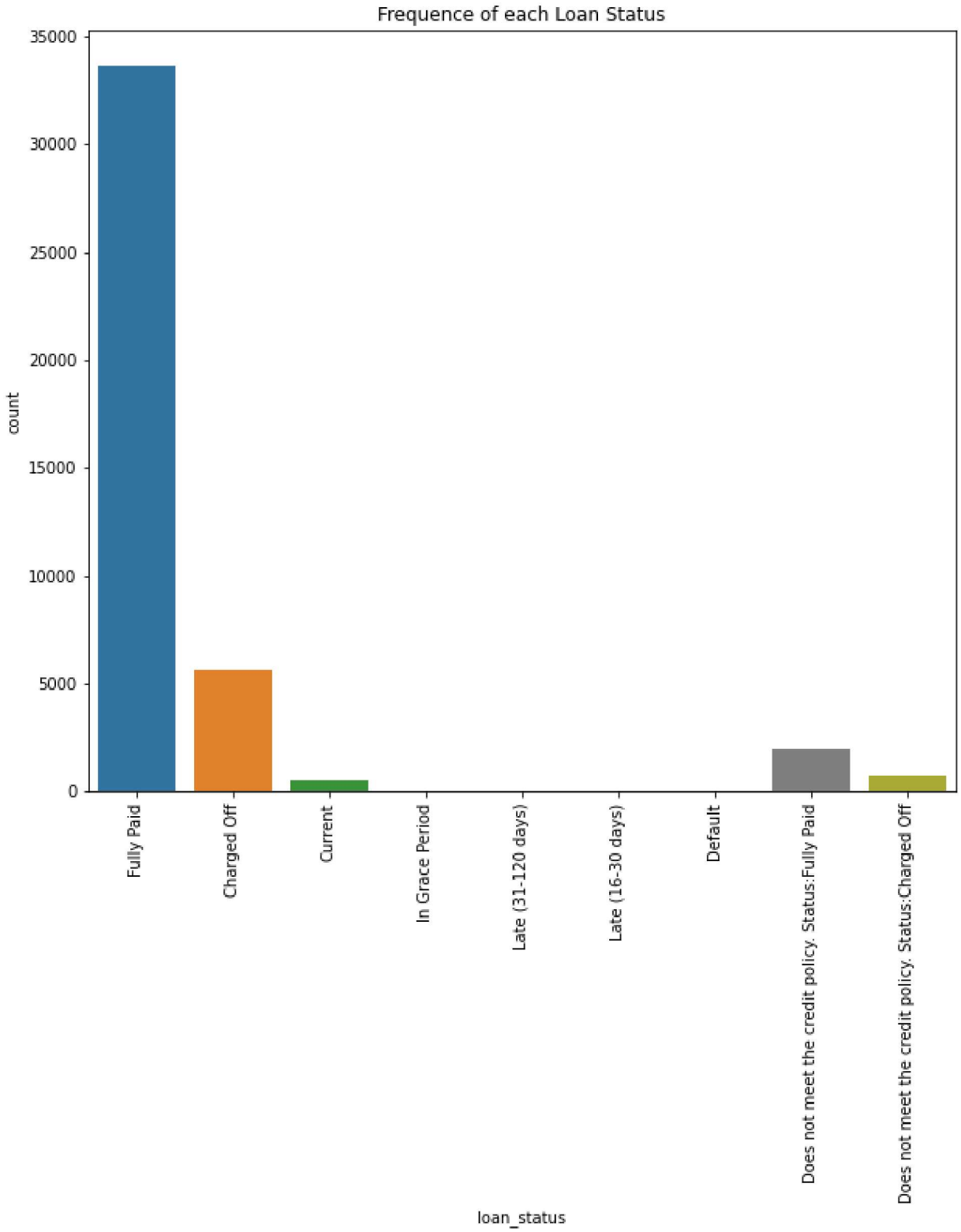
In [148]:
```
df['loan_status'].value_counts()
```

Out[148]:
```
Fully Paid                                            33585
Charged Off                                            5652
Does not meet the credit policy. Status:Fully Paid     1961
Does not meet the credit policy. Status:Charged Off     758
Current                                                 513
In Grace Period                                          16
Late (31-120 days)                                       12
Late (16-30 days)                                         5
Default                                                   1
Name: loan_status, dtype: int64
```

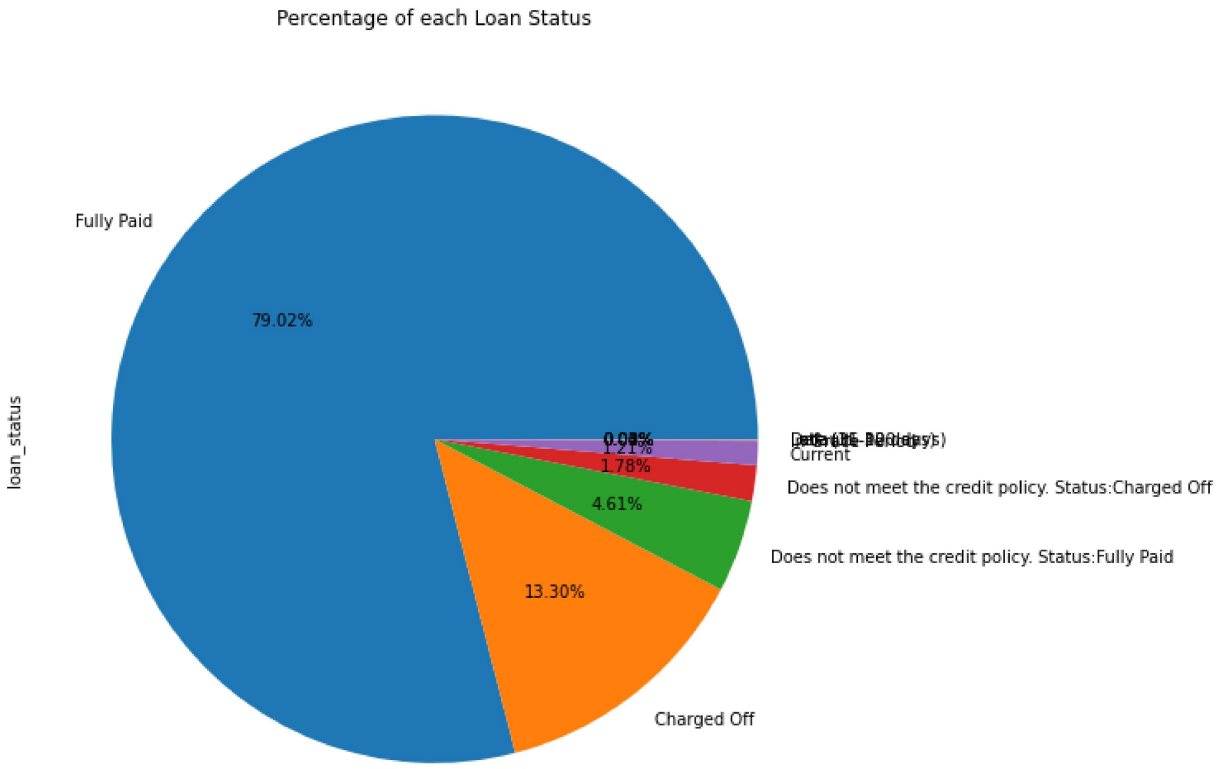
In [ ]:
```

```

In [149]:
```
fig = plt.figure(figsize=(10,9))
sns.countplot(data=df,x='loan_status',).set(title='Frequence of each Loan Status'
plt.xticks(rotation=90)
plt.show()
```



In [150]:
```
fig = plt.figure(figsize=(10,9))
ax = df['loan_status'].value_counts().plot(kind='pie', autopct='%1.2f%%')
ax.set_title('Percentage of each Loan Status')
plt.show()
```



In [ ]:
```
```

In [151]:
```
# $ Machine Learning
```

In [152]:
```
x_df = df.drop(columns=['loan_status'], axis=1)
y_df = df['loan_status']
```

In [153]:
```
x_df
```

Out[153]:

	loan_amnt	term_of_months	int_rate%	installment	grade	emp_length_years	home_owners
0	5000.0	36	10.65	162.87	B	10	RE
1	2500.0	60	15.27	59.83	C	0	RE
2	2400.0	36	15.96	84.33	C	10	RE
3	10000.0	36	13.49	339.31	C	10	RE
4	3000.0	60	12.69	67.79	B	1	RE
...	...	...	...	...	...	...	
42498	5350.0	36	13.12	180.57	D	1	O\
42499	10000.0	36	14.70	345.18	E	1	RE
42500	2000.0	36	7.12	61.87	A	7	MORTGA
42501	6000.0	36	10.59	195.28	C	0	RE
42502	4400.0	36	9.64	141.25	B	2	MORTGA

42503 rows × 30 columns

In [154]:
```
y_df
```

Out[154]:
```
0                                                  Fully Paid
1                                                 Charged Off
2                                                  Fully Paid
3                                                  Fully Paid
4                                                     Current
                                ...
42498    Does not meet the credit policy. Status:Fully ...
42499    Does not meet the credit policy. Status:Fully ...
42500    Does not meet the credit policy. Status:Fully ...
42501    Does not meet the credit policy. Status:Fully ...
42502    Does not meet the credit policy. Status:Fully ...
Name: loan_status, Length: 42503, dtype: object
```

In [155]:
```
df = df[(df['loan_status']=='Fully Paid') | (df['loan_status']=='Charged Off')]
```

In [156]:
```
x_df.shape,y_df.shape
```

Out[156]:
```
((42503, 30), (42503,))
```

In [157]:
```
x_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42503 entries, 0 to 42502
Data columns (total 30 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   loan_amnt             42503 non-null  object
 1   term_of_months        42503 non-null  int64
 2   int_rate%             42503 non-null  object
 3   installment           42503 non-null  float64
 4   grade                 42503 non-null  object
 5   emp_length_years      42503 non-null  object
 6   home_ownership        42503 non-null  object
 7   annual_inc            42503 non-null  float64
 8   verification_status   42503 non-null  object
 9   pymnt_plan            42503 non-null  object
 10  dti                   42503 non-null  float64
 11  delinq_2yrs           42503 non-null  float64
 12  earliest_cr_line      42503 non-null  object
 13  fico_range_low        42503 non-null  float64
```

In [158]:
```
cat_var=['grade','home_ownership','verification_status','application_type']
```

In [159]:
```
for i in cat_var:
    cat_list = 'i'+i
    cat_list= pd.get_dummies(df[i],prefix=i)
    df1=df.join(cat_list)
    df=df1
```

Machine Learning

In [160]:
```
df1
```

Out[160]:

	loan_amnt	term_of_months	int_rate%	installment	grade	emp_length_years	home_owners
0	5000.0	36	10.65	162.87	B	10	RE
1	2500.0	60	15.27	59.83	C	0	RE
2	2400.0	36	15.96	84.33	C	10	RE
3	10000.0	36	13.49	339.31	C	10	RE
5	5000.0	36	7.90	156.46	A	3	RE
...	...	...	...	...	...	...	
39779	2500.0	36	8.07	78.42	A	4	MORTGA
39780	8500.0	36	10.28	275.38	C	3	RE
39781	5000.0	36	8.07	156.84	A	0	MORTGA
39782	5000.0	36	7.43	155.38	A	0	MORTGA
39783	7500.0	36	13.75	255.43	E	0	O\

39237 rows × 47 columns

In [161]:
```
df1.shape
```

Out[161]:
```
(39237, 47)
```

In [162]:
```
df.shape
```

Out[162]:
```
(39237, 47)
```

In [163]:
```
df.sample(5)
```

Out[163]:

	loan_amnt	term_of_months	int_rate%	installment	grade	emp_length_years	home_owners
2240	30000.0	60	12.42	673.72	B	0	RE
37702	20000.0	36	14.74	690.74	D	0	MORTGA
4822	15000.0	60	9.91	318.05	B	1	MORTGA
15110	10000.0	36	7.49	311.02	A	4	O\
19923	10000.0	36	10.74	326.16	B	4	MORTGA

5 rows × 47 columns

In [164]:
```
df.drop(columns=cat_var,inplace=True)
```

In [165]:
```
df.columns.to_list()
```

Out[165]:
```
['loan_amnt',
 'term_of_months',
 'int_rate%',
 'installment',
 'emp_length_years',
 'annual_inc',
 'loan_status',
 'pymnt_plan',
 'dti',
 'delinq_2yrs',
 'earliest_cr_line',
 'fico_range_low',
 'fico_range_high',
 'inq_last_6mths',
 'open_acc',
 'pub_rec',
 'revol_bal',
 'revol_util',
 'total_acc',
 'initial_list_status',
 'last_credit_pull_d',
 'last_fico_range_high',
 'last_fico_range_low',
 'policy_code',
 'acc_now_delinq',
 'delinq_amnt',
 'pub_rec_bankruptcies',
 'grade_A',
 'grade_B',
 'grade_C',
 'grade_D',
 'grade_E',
 'grade_F',
 'grade_G',
 'home_ownership_MORTGAGE',
 'home_ownership_NONE',
 'home_ownership_OTHER',
 'home_ownership_OWN',
 'home_ownership_RENT',
 'verification_status_Not Verified',
 'verification_status_Source Verified',
 'verification_status_Verified',
 'application_type_INDIVIDUAL']
```

In [166]:
```
x=df.drop(columns=['loan_status','earliest_cr_line'])
```

In [167]:
```
x
```

Out[167]:

	loan_amnt	term_of_months	int_rate%	installment	emp_length_years	annual_inc	pymnt_p
0	5000.0	36	10.65	162.87	10	24000.0	
1	2500.0	60	15.27	59.83	0	30000.0	
2	2400.0	36	15.96	84.33	10	12252.0	
3	10000.0	36	13.49	339.31	10	49200.0	
5	5000.0	36	7.90	156.46	3	36000.0	
...	...	...	...	...	...	...	...
39779	2500.0	36	8.07	78.42	4	110000.0	
39780	8500.0	36	10.28	275.38	3	18000.0	
39781	5000.0	36	8.07	156.84	0	100000.0	
39782	5000.0	36	7.43	155.38	0	200000.0	
39783	7500.0	36	13.75	255.43	0	22000.0	

39237 rows × 41 columns

In [168]:
```
y=df['loan_status']
```

In [169]:
```
y
```

Out[169]:
```
0         Fully Paid
1        Charged Off
2         Fully Paid
3         Fully Paid
5         Fully Paid
             ...
39779     Fully Paid
39780     Fully Paid
39781     Fully Paid
39782     Fully Paid
39783     Fully Paid
Name: loan_status, Length: 39237, dtype: object
```

In [170]:
```
from sklearn.model_selection import train_test_split
```

In [171]:
```
import category_encoders as ce
```

In [172]:
```
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size=0.3)
```

In [173]:
```
X_train
```

Out[173]:

	loan_amnt	term_of_months	int_rate%	installment	emp_length_years	annual_inc	pymn
32531	1000.0	36	7.88	31.29	3	72000.0	
7032	4200.0	36	7.51	130.67	0	40000.0	
14260	8400.0	36	11.99	278.97	2	85000.0	
9781	8000.0	36	11.99	265.68	7	100800.0	
31956	5500.0	36	10.99	180.05	1	30000.0	
...	...	...	...	...	...	...	
14013	15000.0	36	14.79	518.44	1 <	45000.0	
29936	10000.0	36	7.88	312.82	4	34995.0	
23496	4000.0	36	12.98	134.74	0	32400.0	
31775	5000.0	36	14.96	173.24	10	60000.0	
32212	10000.0	36	15.33	348.29	10	83628.0	

27465 rows × 41 columns

In [174]:
```
Y_train
```

Out[174]:
```
32531    Fully Paid
7032     Fully Paid
14260    Fully Paid
9781     Fully Paid
31956    Fully Paid
            ...
14013    Fully Paid
29936    Fully Paid
23496    Fully Paid
31775    Fully Paid
32212    Fully Paid
Name: loan_status, Length: 27465, dtype: object
```

In [175]:
```
X_test
```

Out[175]:

	loan_amnt	term_of_months	int_rate%	installment	emp_length_years	annual_inc	pymnt_p
36736	5000.0	36	8.00	156.69	3	119000.00	
20763	10000.0	36	10.74	326.16	10	100000.00	
10728	21225.0	60	12.99	482.83	10	64043.28	
4951	15075.0	36	6.03	458.82	8	62000.00	
19689	20000.0	60	13.06	455.68	7	120000.00	
...	...	...	...	...	...	...	
13296	6000.0	36	10.99	196.41	10	141000.00	
33523	1000.0	36	8.94	31.78	2	780000.00	
25508	10000.0	36	15.95	351.33	4	71028.00	
8173	4000.0	36	13.49	135.73	5	37200.00	
33555	5500.0	36	11.48	181.33	4	14400.00	

11772 rows × 41 columns

In [176]:
```
Y_test
```

Out[176]:
```
36736     Fully Paid
20763     Fully Paid
10728     Fully Paid
4951      Fully Paid
19689     Fully Paid
             ...
13296     Fully Paid
33523     Fully Paid
25508    Charged Off
8173      Fully Paid
33555     Fully Paid
Name: loan_status, Length: 11772, dtype: object
```

In [177]:
```
X_train.columns
```

Out[177]:
```
Index(['loan_amnt', 'term_of_months', 'int_rate%', 'installment',
       'emp_length_years', 'annual_inc', 'pymnt_plan', 'dti', 'delinq_2yrs',
       'fico_range_low', 'fico_range_high', 'inq_last_6mths', 'open_acc',
       'pub_rec', 'revol_bal', 'revol_util', 'total_acc',
       'initial_list_status', 'last_credit_pull_d', 'last_fico_range_high',
       'last_fico_range_low', 'policy_code', 'acc_now_delinq', 'delinq_amnt',
       'pub_rec_bankruptcies', 'grade_A', 'grade_B', 'grade_C', 'grade_D',
       'grade_E', 'grade_F', 'grade_G', 'home_ownership_MORTGAGE',
       'home_ownership_NONE', 'home_ownership_OTHER', 'home_ownership_OWN',
       'home_ownership_RENT', 'verification_status_Not Verified',
       'verification_status_Source Verified', 'verification_status_Verified',
       'application_type_INDIVIDUAL'],
      dtype='object')
```

In [178]:
```
encoder =ce.OrdinalEncoder(cols=[ 'loan_amnt', 'term_of_months', 'int_rate%', 'ir
       'emp_length_years', 'annual_inc', 'pymnt_plan', 'dti', 'delinq_2yrs',
       'fico_range_low', 'fico_range_high', 'inq_last_6mths', 'open_acc',
       'pub_rec', 'revol_bal', 'revol_util', 'total_acc',
       'initial_list_status', 'last_credit_pull_d', 'last_fico_range_high',
       'last_fico_range_low', 'policy_code', 'acc_now_delinq', 'delinq_amnt',
       'pub_rec_bankruptcies', 'grade_A', 'grade_B', 'grade_C', 'grade_D',
       'grade_E', 'grade_F', 'grade_G', 'home_ownership_MORTGAGE',
       'home_ownership_NONE', 'home_ownership_OTHER', 'home_ownership_OWN',
       'home_ownership_RENT', 'verification_status_Not Verified',
       'verification_status_Source Verified', 'verification_status_Verified',
       'application_type_INDIVIDUAL'])
```

In [179]:
```
print (encoder)
```

```
OrdinalEncoder(cols=['loan_amnt', 'term_of_months', 'int_rate%', 'installment',
                     'emp_length_years', 'annual_inc', 'pymnt_plan', 'dti',
                     'delinq_2yrs', 'fico_range_low', 'fico_range_high',
                     'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal',
                     'revol_util', 'total_acc', 'initial_list_status',
                     'last_credit_pull_d', 'last_fico_range_high',
                     'last_fico_range_low', 'policy_code', 'acc_now_delinq',
                     'delinq_amnt', 'pub_rec_bankruptcies', 'grade_A',
                     'grade_B', 'grade_C', 'grade_D', 'grade_E', ...])
```

In [180]:
```
X_train = encoder.fit_transform(X_train)
```

In [181]:
```
X_train
```

Out[181]:

	loan_amnt	term_of_months	int_rate%	installment	emp_length_years	annual_inc	pymnt_p
32531	1	1	1	1	1	1	
7032	2	1	2	2	2	2	
14260	3	1	3	3	3	3	
9781	4	1	3	4	4	4	
31956	5	1	4	5	5	5	
...	...	...	...	...	...	...	
14013	22	1	124	7245	10	9	
29936	17	1	1	357	9	4070	
23496	13	1	167	7682	2	109	
31775	24	1	98	1999	6	6	
32212	17	1	54	286	6	4071	

27465 rows × 41 columns

In [182]:
```
X_test = encoder.fit_transform(X_test)
```

In [183]:
```
X_test
```

Out[183]:

	loan_amnt	term_of_months	int_rate%	installment	emp_length_years	annual_inc	pymnt_p
36736	24.0	1	112.0	885.0	1	46.0	
20763	17.0	1	127.0	2951.0	6	88.0	
10728	-1.0	2	65.0	-1.0	6	-1.0	
4951	598.0	1	11.0	-1.0	12	97.0	
19689	6.0	2	139.0	3427.0	4	36.0	
...	...	...	...	...	...	...	
13296	33.0	1	4.0	3829.0	6	174.0	
33523	1.0	1	13.0	-1.0	3	1394.0	
25508	17.0	1	113.0	4666.0	9	-1.0	
8173	13.0	1	72.0	551.0	7	760.0	
33555	5.0	1	119.0	-1.0	9	196.0	

11772 rows × 41 columns

In [184]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [185]:
```
rfc = RandomForestClassifier(random_state=0)
```

In [186]:
```
rfc.fit(X_train,Y_train)
```

Out[186]:
```
RandomForestClassifier(random_state=0)
```

In [187]:
```
y_pred = rfc.predict(X_test)
```

In [188]:
```
y_pred
```

Out[188]:
```
array(['Fully Paid', 'Fully Paid', 'Charged Off', ..., 'Charged Off',
       'Fully Paid', 'Fully Paid'], dtype=object)
```

In [189]:
```
len(y_pred)
```

Out[189]:
```
11772
```

In [190]:
```
from sklearn.metrics import accuracy_score
```

In [191]:
```
print ('Model  Score is',accuracy_score(Y_test,y_pred))
```

```
Model  Score is 0.9018858307849134
```

In RandomForestClassifier we can see Mode score is 90%

In [192]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [193]:
```
gini_model = DecisionTreeClassifier(criterion='gini',max_depth=3,random_state=0)
```

In [194]:
```
gini_model.fit(X_train,Y_train)
```

Out[194]:
```
DecisionTreeClassifier(max_depth=3, random_state=0)
```

In [195]:
```
print ('Model Score using impurity',accuracy_score(Y_test,y_pred))
```

```
Model Score using impurity 0.9018858307849134
```

In DecisionTreeClassifier Mode score is 0.90 %

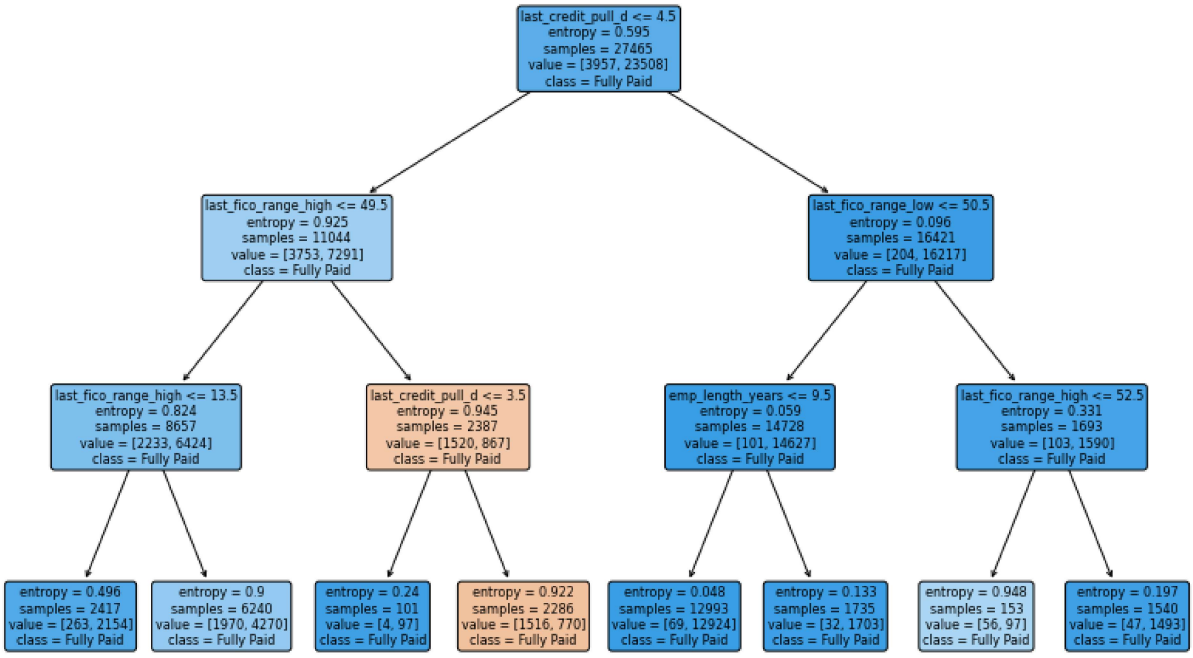
In [196]:
```
from sklearn.tree import plot_tree
```

In [197]:
```
dec_model = DecisionTreeClassifier(criterion='entropy', max_depth=3)

dec_model.fit(X_train, Y_train)

dec_pred = dec_model.predict(X_test)
```

In [198]:
```
plt.figure(figsize=(14,9))
plot_tree(dec_model,
          feature_names=X_test.columns,
          class_names=dec_pred,
          filled=True,
          rounded=True,
          fontsize=8)
plt.show()
```



In [205]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [206]:
```
knn = KNeighborsClassifier(n_neighbors=1)
```

In [207]:
```
knn.fit(X_train,Y_train)
```

Out[207]:
```
KNeighborsClassifier(n_neighbors=1)
```

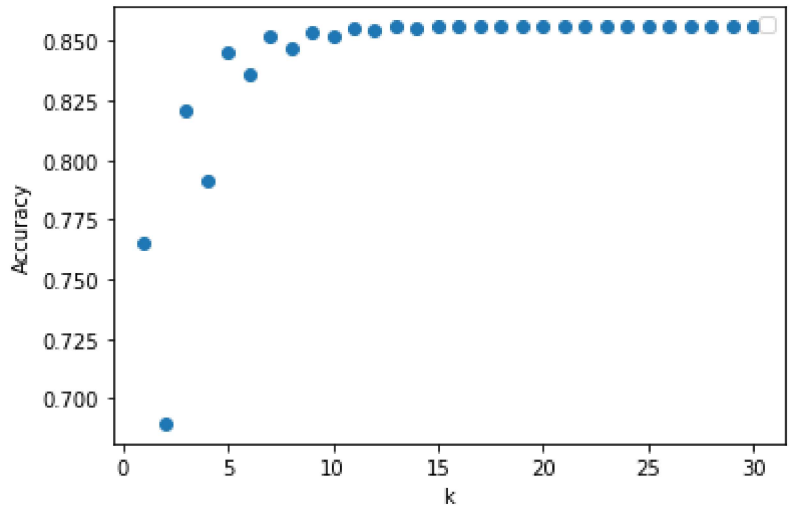
In [208]:
```
knn.score(X_test,Y_test)
```

Out[208]:
```
0.7654604145429833
```

In KNearest Neaighbour Algorithem Model Score is 76 %

In [221]:
```
k_range = range(1,31)
score = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,Y_train)
    score.append(knn.score(X_test,Y_test))
plt.xlabel("k")
plt.ylabel("Accuracy")
plt.legend()
plt.scatter(k_range,score)
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label s
tart with an underscore are ignored when legend() is called with no argument.
```



In Above chart we can Analysis

-- 1) In Above plot shows the high in between 10 to 30 which have Loan Amount is fully Paid

-- 2) It also show's low in between 5 to 10 which have Not Paid there installment

In [ ]: