Lending Club Case Study

In [ ]:

Importing required modules

In [68]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [40]:
```
import warnings
warnings.filterwarnings('ignore')
```

Loading the Data

In [41]:
```
loan_data = pd.read_csv(r"C:\Users\DELL\Downloads\loan.csv")
```

Understanding the Data

In [42]:
```
loan_data.head()
```

Out[42]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	gr
0	1077501	1296599	5000	5000	4975.0	36 months	10.65%	162.87	
1	1077430	1314167	2500	2500	2500.0	60 months	15.27%	59.83	
2	1077175	1313524	2400	2400	2400.0	36 months	15.96%	84.33	
3	1076863	1277178	10000	10000	10000.0	36 months	13.49%	339.31	
4	1075358	1311748	3000	3000	3000.0	60 months	12.69%	67.79	

5 rows × 111 columns

In [43]:
```
loan_data.shape
```

Out[43]:
```
(39717, 111)
```

In [44]:
```
loan_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Columns: 111 entries, id to total_il_high_credit_limit
dtypes: float64(74), int64(13), object(24)
memory usage: 33.6+ MB
```

In [45]:
```
loan_data.describe
```

Out[45]:

```
<bound method NDFrame.describe of             id  member_id  loan_amnt  funded_amnt
funded_amnt_inv  \
0      1077501    1296599       5000         5000           4975.0
1      1077430    1314167       2500         2500           2500.0
2      1077175    1313524       2400         2400           2400.0
3      1076863    1277178      10000        10000          10000.0
4      1075358    1311748       3000         3000           3000.0
...        ...        ...        ...          ...              ...
39712    92187      92174       2500         2500           1075.0
39713    90665      90607       8500         8500            875.0
39714    90395      90390       5000         5000           1325.0
39715    90376      89243       5000         5000            650.0
39716    87023      86999       7500         7500            800.0

             term int_rate  installment grade sub_grade  ...  \
0       36 months   10.65%       162.87     B        B2  ...
1       60 months   15.27%        59.83     C        C4  ...
2       36 months   15.96%        84.33     C        C5  ...
3       36 months   13.49%       339.31     C        C1  ...
4       60 months   12.69%        67.79     B        B5  ...
...           ...      ...          ...   ...       ...  ...
39712   36 months    8.07%        78.42     A        A4  ...
39713   36 months   10.28%       275.38     C        C1  ...
39714   36 months    8.07%       156.84     A        A4  ...
39715   36 months    7.43%       155.38     A        A2  ...
39716   36 months   13.75%       255.43     E        E2  ...

      num_tl_90g_dpd_24m num_tl_op_past_12m pct_tl_nvr_dlq  percent_bc_gt_75  \
0                    NaN                NaN            NaN               NaN
1                    NaN                NaN            NaN               NaN
2                    NaN                NaN            NaN               NaN
3                    NaN                NaN            NaN               NaN
4                    NaN                NaN            NaN               NaN
...                  ...                ...            ...               ...
39712                NaN                NaN            NaN               NaN
39713                NaN                NaN            NaN               NaN
39714                NaN                NaN            NaN               NaN
39715                NaN                NaN            NaN               NaN
39716                NaN                NaN            NaN               NaN

      pub_rec_bankruptcies tax_liens tot_hi_cred_lim total_bal_ex_mort  \
0                      0.0       0.0             NaN               NaN
1                      0.0       0.0             NaN               NaN
2                      0.0       0.0             NaN               NaN
3                      0.0       0.0             NaN               NaN
4                      0.0       0.0             NaN               NaN
...                    ...       ...             ...               ...
39712                  NaN       NaN             NaN               NaN
39713                  NaN       NaN             NaN               NaN
39714                  NaN       NaN             NaN               NaN
39715                  NaN       NaN             NaN               NaN
39716                  NaN       NaN             NaN               NaN

      total_bc_limit total_il_high_credit_limit
0                NaN                       NaN
1                NaN                       NaN
2                NaN                       NaN
3                NaN                       NaN
4                NaN                       NaN
...              ...                       ...
```

```
39712            NaN                        NaN
39713            NaN                        NaN
39714            NaN                        NaN
39715            NaN                        NaN
39716            NaN                        NaN

[39717 rows x 111 columns]>
```

OBSERVATION:We can observe there are a lot of coloumns with NaN values.

In [46]:
```
loan_data.isnull().sum() * 100 / len(loan_data)
```

Out[46]:

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
mths_since_last_record	92.985372
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
next_pymnt_d	97.129693
last_credit_pull_d	0.005036
collections_12_mths_ex_med	0.140998
mths_since_last_major_derog	100.000000
policy_code	0.000000
application_type	0.000000
annual_inc_joint	100.000000
dti_joint	100.000000
verification_status_joint	100.000000
acc_now_delinq	0.000000
tot_coll_amt	100.000000
tot_cur_bal	100.000000
open_acc_6m	100.000000

```
open_il_6m                      100.000000
open_il_12m                     100.000000
open_il_24m                     100.000000
mths_since_rcnt_il              100.000000
total_bal_il                    100.000000
il_util                         100.000000
open_rv_12m                     100.000000
open_rv_24m                     100.000000
max_bal_bc                      100.000000
all_util                        100.000000
total_rev_hi_lim                100.000000
inq_fi                          100.000000
total_cu_tl                     100.000000
inq_last_12m                    100.000000
acc_open_past_24mths            100.000000
avg_cur_bal                     100.000000
bc_open_to_buy                  100.000000
bc_util                         100.000000
chargeoff_within_12_mths          0.140998
delinq_amnt                       0.000000
mo_sin_old_il_acct              100.000000
mo_sin_old_rev_tl_op            100.000000
mo_sin_rcnt_rev_tl_op           100.000000
mo_sin_rcnt_tl                  100.000000
mort_acc                        100.000000
mths_since_recent_bc            100.000000
mths_since_recent_bc_dlq        100.000000
mths_since_recent_inq           100.000000
mths_since_recent_revol_delinq  100.000000
num_accts_ever_120_pd           100.000000
num_actv_bc_tl                  100.000000
num_actv_rev_tl                 100.000000
num_bc_sats                     100.000000
num_bc_tl                       100.000000
num_il_tl                       100.000000
num_op_rev_tl                   100.000000
num_rev_accts                   100.000000
num_rev_tl_bal_gt_0             100.000000
num_sats                        100.000000
num_tl_120dpd_2m                100.000000
num_tl_30dpd                    100.000000
num_tl_90g_dpd_24m              100.000000
num_tl_op_past_12m              100.000000
pct_tl_nvr_dlq                  100.000000
percent_bc_gt_75                100.000000
pub_rec_bankruptcies              1.754916
tax_liens                         0.098195
tot_hi_cred_lim                 100.000000
total_bal_ex_mort               100.000000
total_bc_limit                  100.000000
total_il_high_credit_limit      100.000000
dtype: float64
```

OBSERVATION: We can see there are a lot of coloumns with 100% null values. These can be dropped for our analysis.

In [47]:
```
print(loan_data.nunique(axis=0))
```

id	39717
member_id	39717
loan_amnt	885
funded_amnt	1041
funded_amnt_inv	8205
term	2
int_rate	371
installment	15383
grade	7
sub_grade	35
emp_title	28820
emp_length	11
home_ownership	5
annual_inc	5318
verification_status	3
issue_d	55
loan_status	3
pymnt_plan	1
url	39717
desc	26527
purpose	14
title	19615
zip_code	823
addr_state	50
dti	2868
delinq_2yrs	11
earliest_cr_line	526
inq_last_6mths	9
mths_since_last_delinq	95
mths_since_last_record	111
open_acc	40
pub_rec	5
revol_bal	21711
revol_util	1089
total_acc	82
initial_list_status	1
out_prncp	1137
out_prncp_inv	1138
total_pymnt	37850
total_pymnt_inv	37518
total_rec_prncp	7976
total_rec_int	35148
total_rec_late_fee	1356
recoveries	4040
collection_recovery_fee	2616
last_pymnt_d	101
last_pymnt_amnt	34930
next_pymnt_d	2
last_credit_pull_d	106
collections_12_mths_ex_med	1
mths_since_last_major_derog	0
policy_code	1
application_type	1
annual_inc_joint	0
dti_joint	0
verification_status_joint	0
acc_now_delinq	1
tot_coll_amt	0
tot_cur_bal	0
open_acc_6m	0

```
open_il_6m                      0
open_il_12m                     0
open_il_24m                     0
mths_since_rcnt_il              0
total_bal_il                    0
il_util                         0
open_rv_12m                     0
open_rv_24m                     0
max_bal_bc                      0
all_util                        0
total_rev_hi_lim                0
inq_fi                          0
total_cu_tl                     0
inq_last_12m                    0
acc_open_past_24mths            0
avg_cur_bal                     0
bc_open_to_buy                  0
bc_util                         0
chargeoff_within_12_mths        1
delinq_amnt                     1
mo_sin_old_il_acct              0
mo_sin_old_rev_tl_op            0
mo_sin_rcnt_rev_tl_op           0
mo_sin_rcnt_tl                  0
mort_acc                        0
mths_since_recent_bc            0
mths_since_recent_bc_dlq        0
mths_since_recent_inq           0
mths_since_recent_revol_delinq  0
num_accts_ever_120_pd           0
num_actv_bc_tl                  0
num_actv_rev_tl                 0
num_bc_sats                     0
num_bc_tl                       0
num_il_tl                       0
num_op_rev_tl                   0
num_rev_accts                   0
num_rev_tl_bal_gt_0             0
num_sats                        0
num_tl_120dpd_2m                0
num_tl_30dpd                    0
num_tl_90g_dpd_24m              0
num_tl_op_past_12m              0
pct_tl_nvr_dlq                  0
percent_bc_gt_75                0
pub_rec_bankruptcies            3
tax_liens                       1
tot_hi_cred_lim                 0
total_bal_ex_mort               0
total_bc_limit                  0
total_il_high_credit_limit      0
dtype: int64
```

OBSERVATION: There are many coloumns with 1 uniques value. These can also be dropped as they are not useful for our analysis.

Data Cleaning

In [48]:
```
#Removing coloumns with 100% null values
loan_data = loan_data.dropna(axis=1, how='all')
```

In [49]:
```
loan_data.shape
```

Out[49]:
```
(39717, 57)
```

In [50]:
```
#Removing coloumns with more than 30% null values
loan_data = loan_data.dropna(thresh=loan_data.shape[0]*0.7,axis=1)
```

In [51]:
```
loan_data.shape
```

Out[51]:
```
(39717, 53)
```

In [52]:
```
#checking for unique values
print(loan_data.nunique(axis=0))
```

```
id                            39717
member_id                     39717
loan_amnt                       885
funded_amnt                    1041
funded_amnt_inv                8205
term                              2
int_rate                        371
installment                   15383
grade                             7
sub_grade                        35
emp_title                     28820
emp_length                       11
home_ownership                    5
annual_inc                     5318
verification_status               3
issue_d                          55
loan_status                       3
pymnt_plan                        1
url                           39717
purpose                          14
title                         19615
zip_code                        823
addr_state                       50
dti                            2868
delinq_2yrs                      11
earliest_cr_line                526
inq_last_6mths                    9
open_acc                         40
pub_rec                           5
revol_bal                     21711
revol_util                     1089
total_acc                        82
initial_list_status               1
out_prncp                      1137
out_prncp_inv                  1138
total_pymnt                   37850
total_pymnt_inv               37518
total_rec_prncp                7976
total_rec_int                 35148
total_rec_late_fee             1356
recoveries                     4040
collection_recovery_fee        2616
last_pymnt_d                    101
last_pymnt_amnt               34930
last_credit_pull_d              106
collections_12_mths_ex_med        1
policy_code                       1
application_type                  1
acc_now_delinq                    1
chargeoff_within_12_mths          1
delinq_amnt                       1
pub_rec_bankruptcies              3
tax_liens                         1
dtype: int64
```

In [53]:

```
#Dropping coloumns having only one unique value.
drop_cols = [c for c in list(loan_data) if loan_data[c].nunique() <= 1]
loan_data = loan_data.drop(columns=drop_cols)
```

In [54]:
```
loan_data.shape
```

Out[54]:
```
(39717, 44)
```

In [55]:
```
#checking the datatypes of coloumns
loan_data.dtypes
```

Out[55]:
```
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
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
out_prncp                 float64
out_prncp_inv             float64
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
```

In [56]:
```
loan_data['emp_title'].value_counts()
```

Out[56]:
```
US Army                                134
Bank of America                        109
IBM                                     66
AT&T                                    59
Kaiser Permanente                       56
                                      ...
Community College of Philadelphia        1
AMEC                                     1
lee county sheriff                       1
Bacon County Board of Education          1
Evergreen Center                         1
Name: emp_title, Length: 28820, dtype: int64
```

Determining which coloumns are not required for our analysis

1. id = This is a unique LC assigned ID for the loan listing. This doest not have any any impact on our analysis.
2. member_id = A unique LC assigned Id for the borrower member. This too does not have any impact on our analysis.
3. emp_title = There are a lot of unique values (about 28820) and cannot be generalized for our analysis.
4. issue_d = The month which the loan was funded is also not useful for our analysis.
5. url = URL for the LC page with listing data. This is also not a useful metric
6. title = The loan title provided by the borrower. Purpose of loan is better metrics than the title provided by user.
7. zip_code = The first 3 numbers of the zip code provided by the borrower in the loan application. Also not a useful metrics.
8. arliest_cr_line: The month the borrower's earliest reported credit line was opened. This information might not directly impact the likelihood of default.
9. nq_last_6mths: The number of inquiries in the past 6 months. It might not be a strong predictor of loan default.
10. open_acc: The number of open credit lines. This might not be a strong predictor of default risk.
11. revol_bal: Total credit revolving balance.
12. revol_util: Revolving line utilization rate.
13. total_acc: The total number of credit lines.
14. out_prncp, out_prncp_inv, total_pymnt, total_pymnt_inv, total_rec_prncp, total_rec_int, total_rec_late_fee: These columns provide information about the loan payment history. these are not useful for predicting default at the time of application.
15. recoveries, collection_recovery_fee: These columns provide information about post-charge-off recovery, which may not directly impact the initial loan approval decision.
16. last_pymnt_d, last_pymnt_amnt, last_credit_pull_d: These columns provide information about the last payment and credit pull, which might be more relevant for post-approval monitoring.

In [57]:
```
unwanted_cols = ['id','member_id','title','emp_title','url','zip_code','earliest_cr_li
                 'out_prncp_inv','total_pymnt','total_pymnt_inv','total_rec_prncp','total
                 'last_credit_pull_d']
```

In [58]:
```
loan_data = loan_data.drop(unwanted_cols, axis =1)
```

In [59]:
```
loan_data.shape
```

Out[59]:
```
(39717, 20)
```

Dropping records which has "Current" as loan status as these are not useful for the analysis.

In [60]:
```
loan_data = loan_data[loan_data['loan_status'] != "Current"]
```

In [61]:
```
loan_data.shape
```

Out[61]:
```
(38577, 20)
```

In [62]:
```
# Dropping NONE records from home_ownership.
loan_data = loan_data[loan_data['home_ownership'] != 'NONE']
loan_data.shape
```

Out[62]:
```
(38574, 20)
```

In [63]:
```
# Dropping missing records from emp_length.
loan_data = loan_data[~loan_data['emp_length'].isnull()]
loan_data.shape
```

Out[63]:
```
(37541, 20)
```

In [64]:
```
# Dropping missing records from pub_rec_bankruptcies.
loan_data = loan_data[~loan_data['pub_rec_bankruptcies'].isnull()]
loan_data.shape
```

Out[64]:
```
(36847, 20)
```

Preprocessing Data

In [65]:
```
#Converting interest rate to float
loan_data['int_rate'] = loan_data['int_rate'].str.rstrip('%').astype('float')
loan_data.head()
```

Out[65]:

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_l
0	5000	5000	4975.0	36 months	10.65	162.87	B	B2	10+
1	2500	2500	2500.0	60 months	15.27	59.83	C	C4	<
2	2400	2400	2400.0	36 months	15.96	84.33	C	C5	10+
3	10000	10000	10000.0	36 months	13.49	339.31	C	C1	10+
5	5000	5000	5000.0	36 months	7.90	156.46	A	A4	

In [66]:
```
#Replacing 10+ years with 10 years and < 1 year with 0 year in emp_length
loan_data['emp_length'] = loan_data['emp_length'].replace({'10+ years': '10 years', '<
loan_data.head()
```

Out[66]:

	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_l
0	5000	5000	4975.0	36 months	10.65	162.87	B	B2	1(
1	2500	2500	2500.0	60 months	15.27	59.83	C	C4	(
2	2400	2400	2400.0	36 months	15.96	84.33	C	C5	1(
3	10000	10000	10000.0	36 months	13.49	339.31	C	C1	1(
5	5000	5000	5000.0	36 months	7.90	156.46	A	A4	

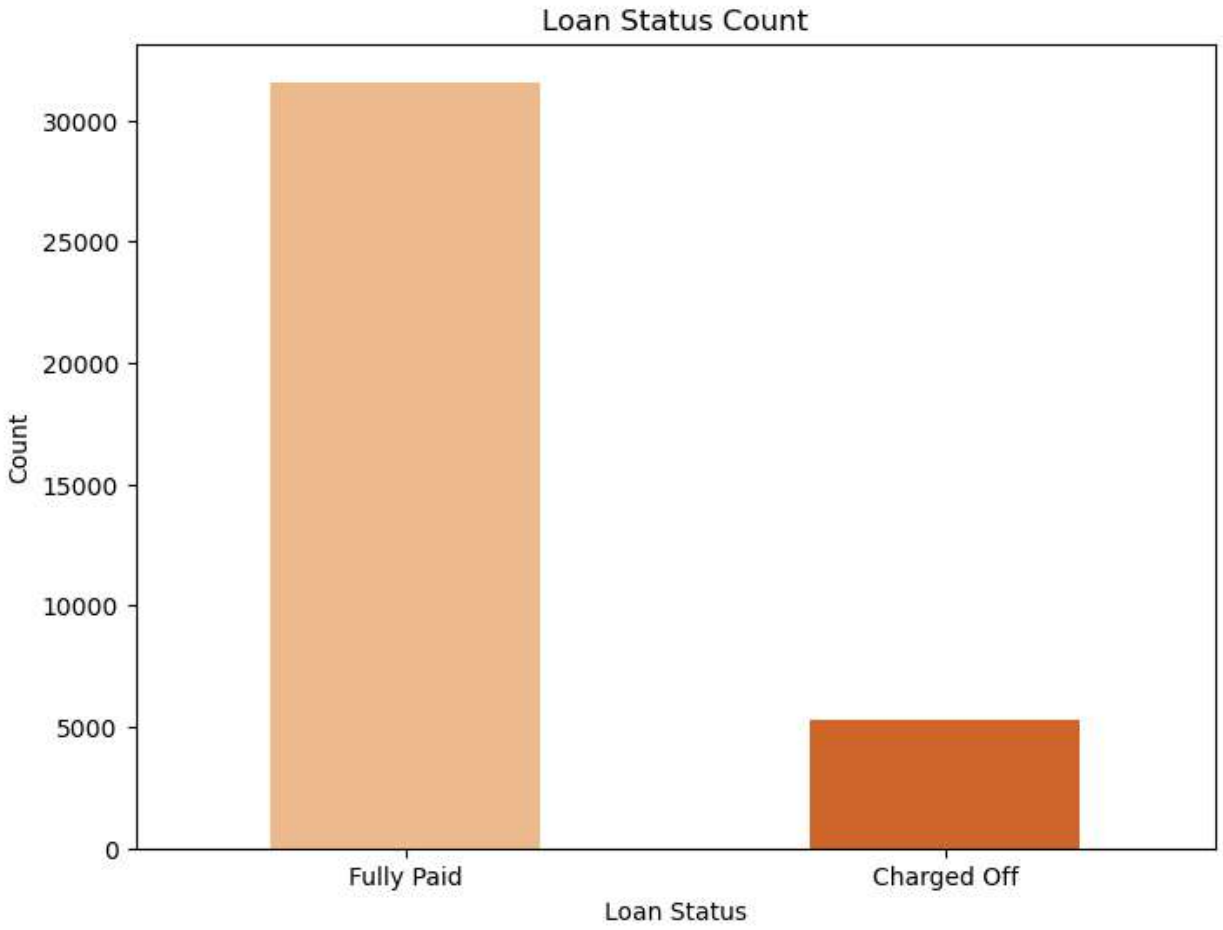
In [67]:
```
#Checking for missing values
loan_data.isnull().sum() * 100 / len(loan_data)
```

Out[67]:
```
loan_amnt               0.0
funded_amnt             0.0
funded_amnt_inv         0.0
term                    0.0
int_rate                0.0
installment             0.0
grade                   0.0
sub_grade               0.0
emp_length              0.0
home_ownership          0.0
annual_inc              0.0
verification_status     0.0
issue_d                 0.0
loan_status             0.0
purpose                 0.0
addr_state              0.0
dti                     0.0
delinq_2yrs             0.0
pub_rec                 0.0
pub_rec_bankruptcies    0.0
dtype: float64
```

Data Analysis

Univariate Analysis

In [79]:
```
#Visualizing the count of Charged off and Fully Paid
plt.figure(figsize=(8, 6))
sns.countplot(x='loan_status', data=loan_data, palette='Oranges',width=0.5)
plt.title('Loan Status Count')
plt.xlabel('Loan Status')
plt.ylabel('Count')
plt.show()
```

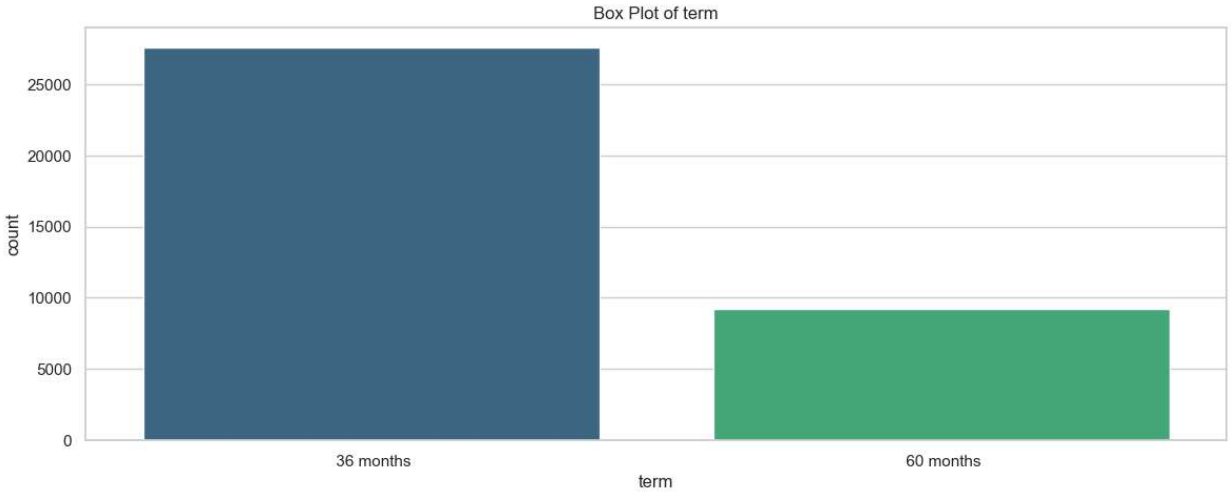


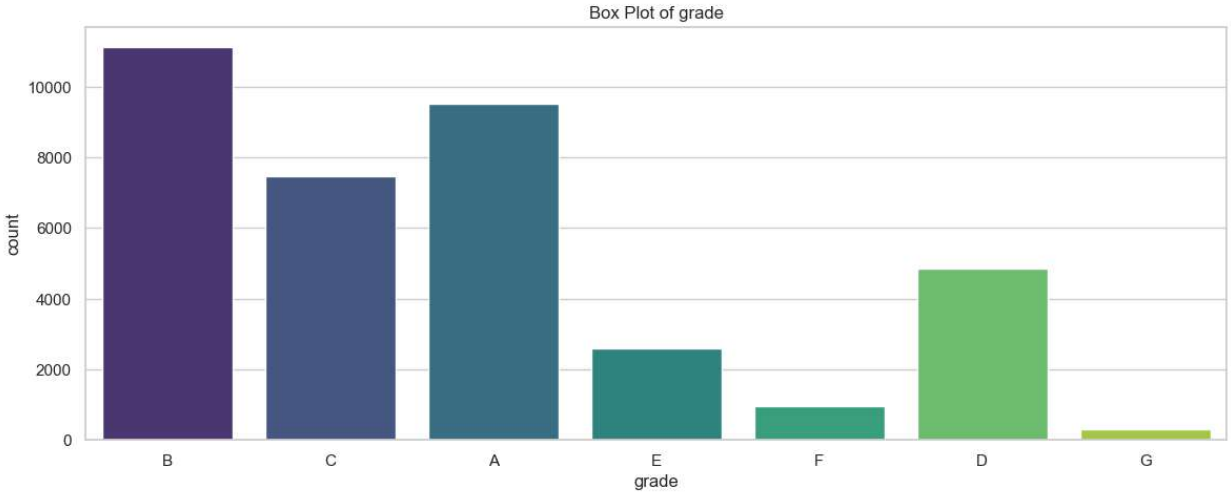
In [82]:
```
status_counts = loan_data['loan_status'].value_counts()
print(status_counts['Charged Off'] / len(loan_data)* 100)
```

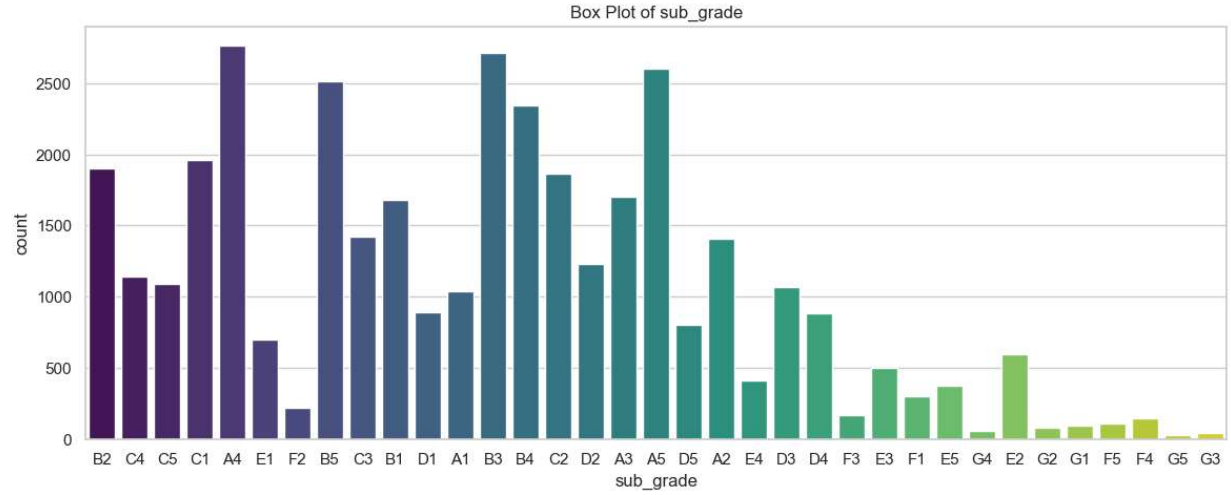
```
14.332238716856189
```

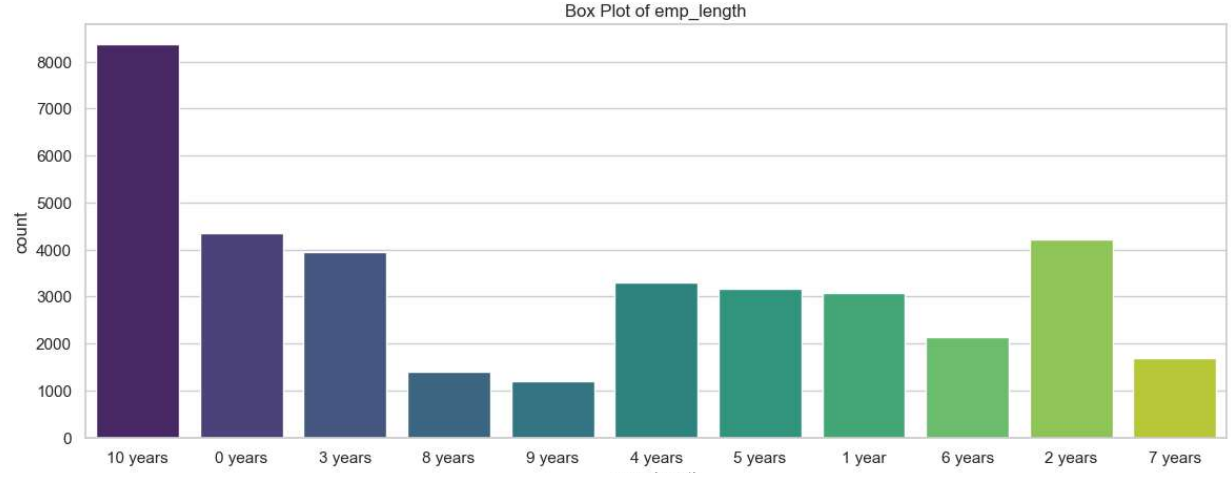
OBSERVATION : About 14.33% loans were charged off.

In [99]:
```
sns.set(style="whitegrid")

categorical_vars = ['term', 'grade', 'sub_grade','emp_length','home_ownership','verifi


fig, axes = plt.subplots(nrows=len(categorical_vars), ncols=1, figsize=(14, 6 * len(ca


for i, cat_var in enumerate(categorical_vars):
    sns.countplot(x=cat_var, data=loan_data, ax=axes[i], palette='viridis')
    axes[i].set_title(f'Box Plot of {cat_var}')



plt.show()
```


Box Plot of term
count
25000
20000
15000
10000
5000
0
36 months
60 months
term

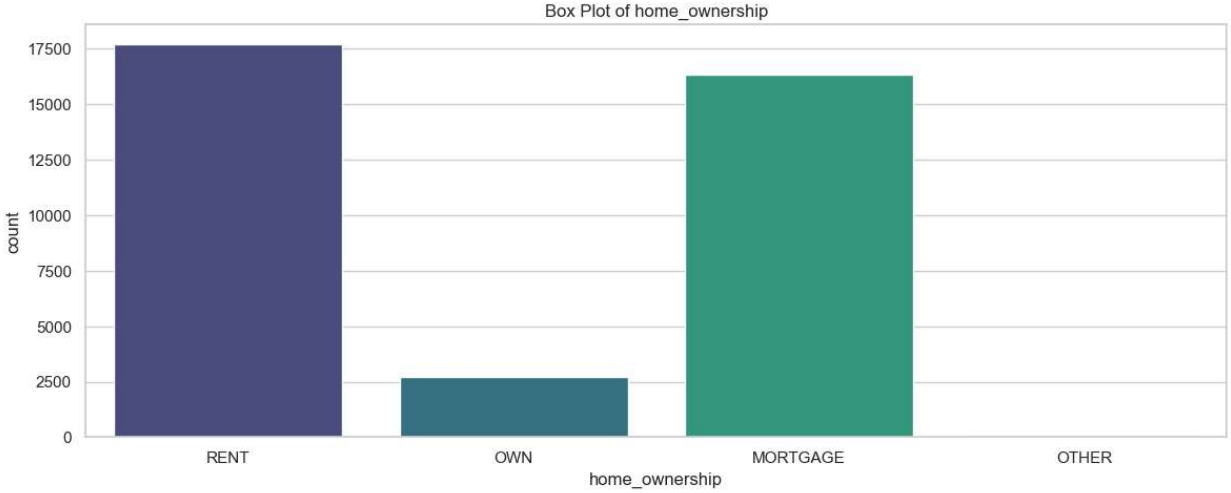

Box Plot of grade
count
10000
8000
6000
4000
2000
0
B
C
A
E
F
D
G
grade


Box Plot of sub_grade
count
2500
2000
1500
1000
500
0
B2 C4 C5 C1 A4 E1 F2 B5 C3 B1 D1 A1 B3 B4 C2 D2 A3 A5 D5 A2 E4 D3 D4 F3 E3 F1 E5 G4 E2 G2 G1 F5 F4 G5 G3
sub_grade

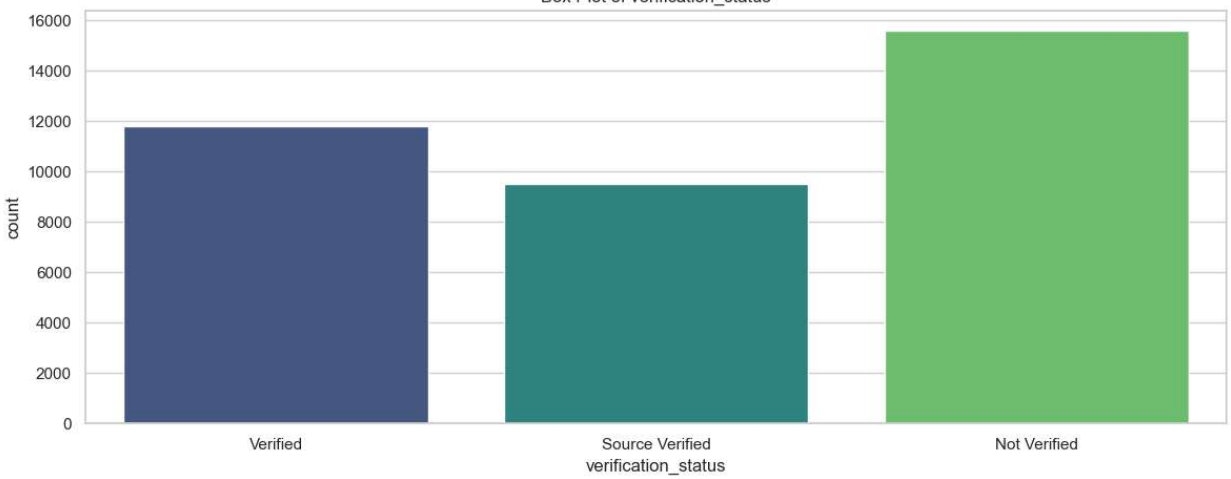

Box Plot of emp_length
count
8000
7000
6000
5000
4000
3000
2000
1000
0
10 years
0 years
3 years
8 years
9 years
4 years
5 years
1 year
6 years
2 years
7 years

emp_length

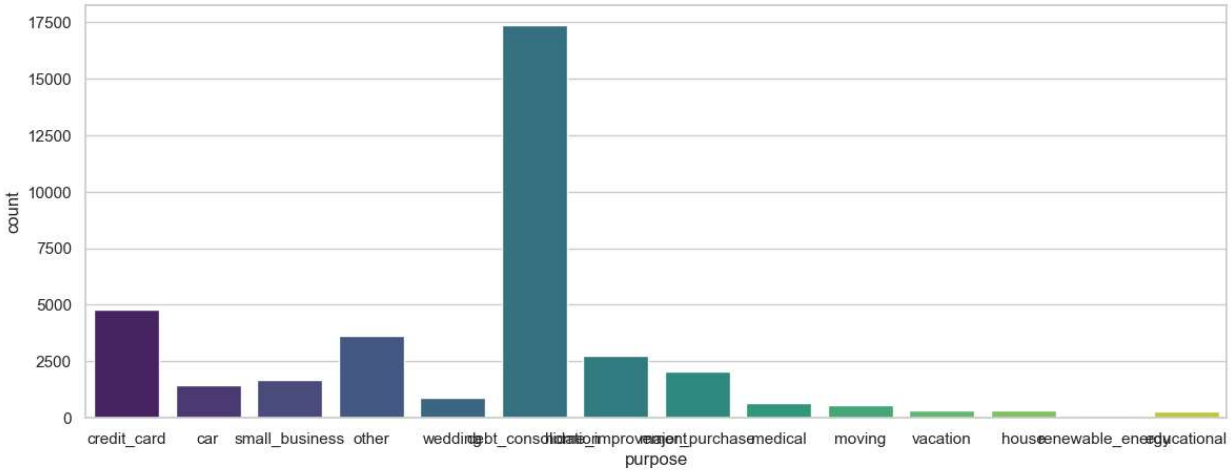
Box Plot of home_ownership



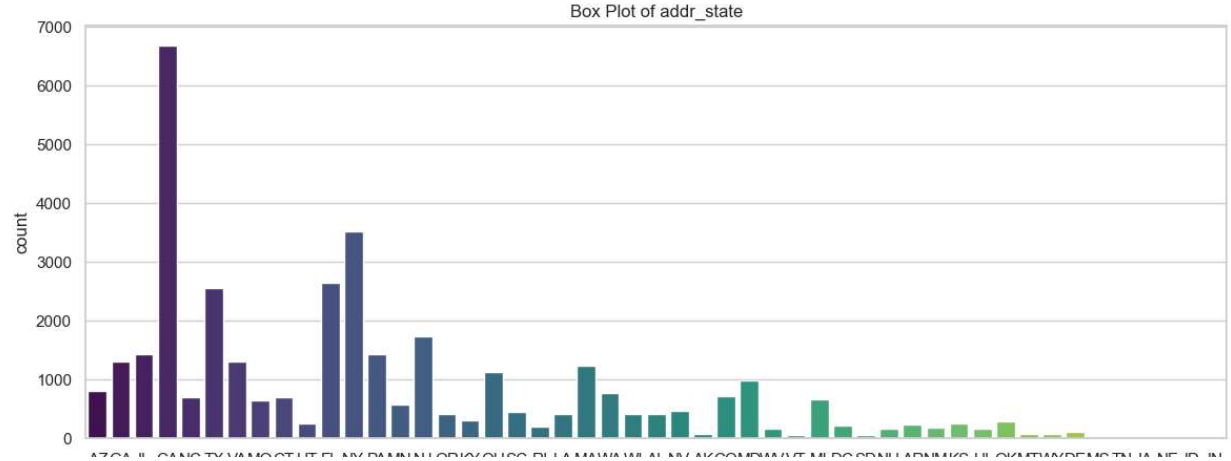
Box Plot of verification_status



Box Plot of purpose



Box Plot of addr_state



AZGA IL CANC TX VAMO CT UT FL NY PAMN NJ OR KY OH SC RI LA MA WA WI AL NV AK COMD WV VT MI DC SD NH AR NM MS HI OK MT WY DE ME IN IA NE ID IN
addr_state

OBSERVATIONS:

1. most of the loans are taken for 36 months.
2. most number of loans are taken in Grade B and least in Grade G
3. most loans are taken by people with employment length greater tha or equal to 10 years.
4. people living on rent are taking most loans.
5. Most of the loans have status Not verified.
6. debt_consolidation is major reason for taking loans.
7. People of California have taken most loans.

In [ ]: