bankdata-largedata-notebook

March 2, 2025

1 This is about Bankdata Cleansing with PySpark

[1]:
```
#csv_df = spark.read.format('csv').option('header', True).option('inferSchema',␣
↪True).option('escape', '"').load('abfss://
↪65ca3b95-e765-425e-babe-29ac1e3c086a@onelake.dfs.fabric.microsoft.com/
↪6d842f3b-7d08-4037-b1ec-7443755e8379/Files/accepted_2007_to_2018Q4.csv')
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 3, Finished, Available,␣
↪Finished)
```

[2]:
```
#display(csv_df.select("*").filter(col("id") == 61400928))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 4, Finished, Available,␣
↪Finished)
```

[3]:
```
#csv_df.write.format('delta').saveAsTable('inv_bank_data')
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 5, Finished, Available,␣
↪Finished)
```

[4]:
```
from pyspark.sql.functions import *
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 6, Finished, Available,␣
↪Finished)
```

[5]:
```
df = spark.read.format('delta').load("abfss://
↪65ca3b95-e765-425e-babe-29ac1e3c086a@onelake.dfs.fabric.microsoft.com/
↪6d842f3b-7d08-4037-b1ec-7443755e8379/Tables/
↪inv_bank_data",header=True,inferSchema=True)
display(df.head(2))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 7, Finished, Available,␣
↪Finished)
```

```
SynapseWidget(Synapse.DataFrame, b3613963-aea1-4192-a50d-451f07022d39)
```

[6]:
```
display(df.select("*").filter(col("id") == 61400928)) #.show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 8, Finished, Available,␣
↪Finished)

SynapseWidget(Synapse.DataFrame, e7eee87f-cab4-4f4e-9e3f-d0a34021f341)
```

[7]:
```
display(df.printSchema())
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 9, Finished, Available,␣
↪Finished)

root
 |-- id: string (nullable = true)
 |-- member_id: string (nullable = true)
 |-- loan_amnt: double (nullable = true)
 |-- funded_amnt: double (nullable = true)
 |-- funded_amnt_inv: double (nullable = true)
 |-- term: string (nullable = true)
 |-- int_rate: double (nullable = true)
 |-- installment: double (nullable = true)
 |-- grade: string (nullable = true)
 |-- sub_grade: string (nullable = true)
 |-- emp_title: string (nullable = true)
 |-- emp_length: string (nullable = true)
 |-- home_ownership: string (nullable = true)
 |-- annual_inc: double (nullable = true)
 |-- verification_status: string (nullable = true)
 |-- issue_d: string (nullable = true)
 |-- loan_status: string (nullable = true)
 |-- pymnt_plan: string (nullable = true)
 |-- url: string (nullable = true)
 |-- desc: string (nullable = true)
 |-- purpose: string (nullable = true)
 |-- title: string (nullable = true)
 |-- zip_code: string (nullable = true)
 |-- addr_state: string (nullable = true)
 |-- dti: double (nullable = true)
 |-- delinq_2yrs: double (nullable = true)
 |-- earliest_cr_line: string (nullable = true)
 |-- fico_range_low: double (nullable = true)
 |-- fico_range_high: double (nullable = true)
 |-- inq_last_6mths: double (nullable = true)
 |-- mths_since_last_delinq: double (nullable = true)
 |-- mths_since_last_record: double (nullable = true)
 |-- open_acc: double (nullable = true)
 |-- pub_rec: double (nullable = true)
 |-- revol_bal: double (nullable = true)
 |-- revol_util: double (nullable = true)
 |-- total_acc: double (nullable = true)
 |-- initial_list_status: string (nullable = true)
```

```
|-- out_prncp: double (nullable = true)
|-- out_prncp_inv: double (nullable = true)
|-- total_pymnt: double (nullable = true)
|-- total_pymnt_inv: double (nullable = true)
|-- total_rec_prncp: double (nullable = true)
|-- total_rec_int: double (nullable = true)
|-- total_rec_late_fee: double (nullable = true)
|-- recoveries: double (nullable = true)
|-- collection_recovery_fee: double (nullable = true)
|-- last_pymnt_d: string (nullable = true)
|-- last_pymnt_amnt: double (nullable = true)
|-- next_pymnt_d: string (nullable = true)
|-- last_credit_pull_d: string (nullable = true)
|-- last_fico_range_high: double (nullable = true)
|-- last_fico_range_low: double (nullable = true)
|-- collections_12_mths_ex_med: double (nullable = true)
|-- mths_since_last_major_derog: double (nullable = true)
|-- policy_code: double (nullable = true)
|-- application_type: string (nullable = true)
|-- annual_inc_joint: double (nullable = true)
|-- dti_joint: double (nullable = true)
|-- verification_status_joint: string (nullable = true)
|-- acc_now_delinq: double (nullable = true)
|-- tot_coll_amt: double (nullable = true)
|-- tot_cur_bal: double (nullable = true)
|-- open_acc_6m: double (nullable = true)
|-- open_act_il: double (nullable = true)
|-- open_il_12m: double (nullable = true)
|-- open_il_24m: double (nullable = true)
|-- mths_since_rcnt_il: double (nullable = true)
|-- total_bal_il: double (nullable = true)
|-- il_util: double (nullable = true)
|-- open_rv_12m: double (nullable = true)
|-- open_rv_24m: double (nullable = true)
|-- max_bal_bc: double (nullable = true)
|-- all_util: double (nullable = true)
|-- total_rev_hi_lim: double (nullable = true)
|-- inq_fi: double (nullable = true)
|-- total_cu_tl: double (nullable = true)
|-- inq_last_12m: double (nullable = true)
|-- acc_open_past_24mths: double (nullable = true)
|-- avg_cur_bal: double (nullable = true)
|-- bc_open_to_buy: double (nullable = true)
|-- bc_util: double (nullable = true)
|-- chargeoff_within_12_mths: double (nullable = true)
|-- delinq_amnt: double (nullable = true)
|-- mo_sin_old_il_acct: double (nullable = true)
|-- mo_sin_old_rev_tl_op: double (nullable = true)
```

```
|-- mo_sin_rcnt_rev_tl_op: double (nullable = true)
|-- mo_sin_rcnt_tl: double (nullable = true)
|-- mort_acc: double (nullable = true)
|-- mths_since_recent_bc: double (nullable = true)
|-- mths_since_recent_bc_dlq: double (nullable = true)
|-- mths_since_recent_inq: double (nullable = true)
|-- mths_since_recent_revol_delinq: double (nullable = true)
|-- num_accts_ever_120_pd: double (nullable = true)
|-- num_actv_bc_tl: double (nullable = true)
|-- num_actv_rev_tl: double (nullable = true)
|-- num_bc_sats: double (nullable = true)
|-- num_bc_tl: double (nullable = true)
|-- num_il_tl: double (nullable = true)
|-- num_op_rev_tl: double (nullable = true)
|-- num_rev_accts: double (nullable = true)
|-- num_rev_tl_bal_gt_0: double (nullable = true)
|-- num_sats: double (nullable = true)
|-- num_tl_120dpd_2m: double (nullable = true)
|-- num_tl_30dpd: double (nullable = true)
|-- num_tl_90g_dpd_24m: double (nullable = true)
|-- num_tl_op_past_12m: double (nullable = true)
|-- pct_tl_nvr_dlq: double (nullable = true)
|-- percent_bc_gt_75: double (nullable = true)
|-- pub_rec_bankruptcies: double (nullable = true)
|-- tax_liens: double (nullable = true)
|-- tot_hi_cred_lim: double (nullable = true)
|-- total_bal_ex_mort: double (nullable = true)
|-- total_bc_limit: double (nullable = true)
|-- total_il_high_credit_limit: double (nullable = true)
|-- revol_bal_joint: double (nullable = true)
|-- sec_app_fico_range_low: double (nullable = true)
|-- sec_app_fico_range_high: double (nullable = true)
|-- sec_app_earliest_cr_line: string (nullable = true)
|-- sec_app_inq_last_6mths: double (nullable = true)
|-- sec_app_mort_acc: double (nullable = true)
|-- sec_app_open_acc: double (nullable = true)
|-- sec_app_revol_util: double (nullable = true)
|-- sec_app_open_act_il: double (nullable = true)
|-- sec_app_num_rev_accts: double (nullable = true)
|-- sec_app_chargeoff_within_12_mths: double (nullable = true)
|-- sec_app_collections_12_mths_ex_med: double (nullable = true)
|-- sec_app_mths_since_last_major_derog: double (nullable = true)
|-- hardship_flag: string (nullable = true)
|-- hardship_type: string (nullable = true)
|-- hardship_reason: string (nullable = true)
|-- hardship_status: string (nullable = true)
|-- deferral_term: double (nullable = true)
|-- hardship_amount: double (nullable = true)
```

```
 |-- hardship_start_date: string (nullable = true)
 |-- hardship_end_date: string (nullable = true)
 |-- payment_plan_start_date: string (nullable = true)
 |-- hardship_length: double (nullable = true)
 |-- hardship_dpd: double (nullable = true)
 |-- hardship_loan_status: string (nullable = true)
 |-- orig_projected_additional_accrued_interest: double (nullable = true)
 |-- hardship_payoff_balance_amount: double (nullable = true)
 |-- hardship_last_payment_amount: double (nullable = true)
 |-- disbursement_method: string (nullable = true)
 |-- debt_settlement_flag: string (nullable = true)
 |-- debt_settlement_flag_date: string (nullable = true)
 |-- settlement_status: string (nullable = true)
 |-- settlement_date: string (nullable = true)
 |-- settlement_amount: double (nullable = true)
 |-- settlement_percentage: double (nullable = true)
 |-- settlement_term: double (nullable = true)
```

[8]:
```
list(df.columns)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 10, Finished, Available,␣
 ↪Finished)
```

[8]:
```
['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',
 'desc',
 'purpose',
 'title',
 'zip_code',
```

```
'addr_state',
'dti',
'delinq_2yrs',
'earliest_cr_line',
'fico_range_low',
'fico_range_high',
'inq_last_6mths',
'mths_since_last_delinq',
'mths_since_last_record',
'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_pymnt_d',
'last_pymnt_amnt',
'next_pymnt_d',
'last_credit_pull_d',
'last_fico_range_high',
'last_fico_range_low',
'collections_12_mths_ex_med',
'mths_since_last_major_derog',
'policy_code',
'application_type',
'annual_inc_joint',
'dti_joint',
'verification_status_joint',
'acc_now_delinq',
'tot_coll_amt',
'tot_cur_bal',
'open_acc_6m',
'open_act_il',
'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',
'chargeoff_within_12_mths',
'delinq_amnt',
'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',
'pub_rec_bankruptcies',
'tax_liens',
'tot_hi_cred_lim',
'total_bal_ex_mort',
'total_bc_limit',
'total_il_high_credit_limit',
'revol_bal_joint',
'sec_app_fico_range_low',
```

```
 'sec_app_fico_range_high',
 'sec_app_earliest_cr_line',
 'sec_app_inq_last_6mths',
 'sec_app_mort_acc',
 'sec_app_open_acc',
 'sec_app_revol_util',
 'sec_app_open_act_il',
 'sec_app_num_rev_accts',
 'sec_app_chargeoff_within_12_mths',
 'sec_app_collections_12_mths_ex_med',
 'sec_app_mths_since_last_major_derog',
 'hardship_flag',
 'hardship_type',
 'hardship_reason',
 'hardship_status',
 'deferral_term',
 'hardship_amount',
 'hardship_start_date',
 'hardship_end_date',
 'payment_plan_start_date',
 'hardship_length',
 'hardship_dpd',
 'hardship_loan_status',
 'orig_projected_additional_accrued_interest',
 'hardship_payoff_balance_amount',
 'hardship_last_payment_amount',
 'disbursement_method',
 'debt_settlement_flag',
 'debt_settlement_flag_date',
 'settlement_status',
 'settlement_date',
 'settlement_amount',
 'settlement_percentage',
 'settlement_term']
```

[9]:
```
rows = df.count()
columns= len(df.columns)
print("Rows",rows,",","Columns",columns) #to find the rows and columns
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 11, Finished, Available,␣
 ↪Finished)
```

```
Rows 2260701 , Columns 151
```

[10]:
```
df.filter(col('id').isNull()).count() #to check the missing/null values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 12, Finished, Available,␣
 ↪Finished)
```

[10]:
```
0
```

[11]:
```
df.filter(col('member_id').isNull()).count() #to check the missing/null values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 13, Finished, Available, Finished)
```

[11]:
```
2260701
```

[12]:
```
df.filter(col("loan_amnt").isNull()).count() #to check the missing/null values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 14, Finished, Available, Finished)
```

[12]:
```
33
```

[13]:
```
df.filter(col("funded_amnt").isNull()).count() ##to check the missing/null values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 15, Finished, Available, Finished)
```

[13]:
```
33
```

[14]:
```
def check_missing_values(data,cl):
    a = data.filter(col(cl).isNull()).count()
    b = data.count()
    c= (a/b) * 100
    return c                                        #to check the missing value percentages of each column passing
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 16, Finished, Available, Finished)
```

[15]:
```
check_missing_values(df,'id')
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 17, Finished, Available, Finished)
```

[15]:
```
0.0
```

[16]:
```
check_missing_values(df,'member_id')
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 18, Finished, Available, Finished)
```

[16]:
```
100.0
```

[17]:
```
check_missing_values(df,"loan_amnt")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 19, Finished, Available,␣
↪Finished)
```

[17]:
```
0.0014597242182845054
```

[18]:
```
check_missing_values(df,"funded_amnt")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 20, Finished, Available,␣
↪Finished)
```

[18]:
```
0.0014597242182845054
```

2 to get all the columns missing values at single instance

[19]:
```
#to get all the columns missing values at single instance
def check_miss_value_pctg(data,lst_cl):
    missing_values = {}
    for i in lst_cl:
        a = data.filter(col(i).isNull()).count()
        missing_values[i] = a
    return (missing_values)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 21, Finished, Available,␣
↪Finished)
```

[20]:
```
check_miss_value_pctg(df,df.columns)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 22, Finished, Available,␣
↪Finished)
```

[20]:
```
{'id': 0,
 'member_id': 2260701,
 'loan_amnt': 33,
 'funded_amnt': 33,
 'funded_amnt_inv': 33,
 'term': 33,
 'int_rate': 33,
 'installment': 33,
 'grade': 33,
 'sub_grade': 33,
 'emp_title': 167002,
 'emp_length': 146940,
 'home_ownership': 33,
 'annual_inc': 37,
 'verification_status': 33,
```

```
'issue_d': 33,
'loan_status': 33,
'pymnt_plan': 33,
'url': 33,
'desc': 2134634,
'purpose': 33,
'title': 23358,
'zip_code': 34,
'addr_state': 33,
'dti': 1744,
'delinq_2yrs': 62,
'earliest_cr_line': 62,
'fico_range_low': 33,
'fico_range_high': 33,
'inq_last_6mths': 63,
'mths_since_last_delinq': 1158535,
'mths_since_last_record': 1901545,
'open_acc': 62,
'pub_rec': 62,
'revol_bal': 33,
'revol_util': 1835,
'total_acc': 62,
'initial_list_status': 33,
'out_prncp': 33,
'out_prncp_inv': 33,
'total_pymnt': 33,
'total_pymnt_inv': 33,
'total_rec_prncp': 33,
'total_rec_int': 33,
'total_rec_late_fee': 33,
'recoveries': 33,
'collection_recovery_fee': 33,
'last_pymnt_d': 2460,
'last_pymnt_amnt': 33,
'next_pymnt_d': 1345343,
'last_credit_pull_d': 105,
'last_fico_range_high': 33,
'last_fico_range_low': 33,
'collections_12_mths_ex_med': 178,
'mths_since_last_major_derog': 1679926,
'policy_code': 33,
'application_type': 33,
'annual_inc_joint': 2139991,
'dti_joint': 2139995,
'verification_status_joint': 2144971,
'acc_now_delinq': 62,
'tot_coll_amt': 70309,
```

```
'tot_cur_bal': 70309,
'open_acc_6m': 866163,
'open_act_il': 866162,
'open_il_12m': 866162,
'open_il_24m': 866162,
'mths_since_rcnt_il': 909957,
'total_bal_il': 866162,
'il_util': 1068883,
'open_rv_12m': 866162,
'open_rv_24m': 866162,
'max_bal_bc': 866162,
'all_util': 866381,
'total_rev_hi_lim': 70309,
'inq_fi': 866162,
'total_cu_tl': 866163,
'inq_last_12m': 866163,
'acc_open_past_24mths': 50063,
'avg_cur_bal': 70379,
'bc_open_to_buy': 74968,
'bc_util': 76104,
'chargeoff_within_12_mths': 178,
'delinq_amnt': 62,
'mo_sin_old_il_acct': 139104,
'mo_sin_old_rev_tl_op': 70310,
'mo_sin_rcnt_rev_tl_op': 70310,
'mo_sin_rcnt_tl': 70309,
'mort_acc': 50063,
'mths_since_recent_bc': 73445,
'mths_since_recent_bc_dlq': 1741000,
'mths_since_recent_inq': 295468,
'mths_since_recent_revol_delinq': 1520342,
'num_accts_ever_120_pd': 70309,
'num_actv_bc_tl': 70309,
'num_actv_rev_tl': 70309,
'num_bc_sats': 58623,
'num_bc_tl': 70309,
'num_il_tl': 70309,
'num_op_rev_tl': 70309,
'num_rev_accts': 70310,
'num_rev_tl_bal_gt_0': 70309,
'num_sats': 58623,
'num_tl_120dpd_2m': 153690,
'num_tl_30dpd': 70309,
'num_tl_90g_dpd_24m': 70309,
'num_tl_op_past_12m': 70309,
'pct_tl_nvr_dlq': 70464,
'percent_bc_gt_75': 75412,
```

```
'pub_rec_bankruptcies': 1398,
'tax_liens': 138,
'tot_hi_cred_lim': 70309,
'total_bal_ex_mort': 50063,
'total_bc_limit': 50063,
'total_il_high_credit_limit': 70309,
'revol_bal_joint': 2152681,
'sec_app_fico_range_low': 2152680,
'sec_app_fico_range_high': 2152680,
'sec_app_earliest_cr_line': 2152680,
'sec_app_inq_last_6mths': 2152680,
'sec_app_mort_acc': 2152680,
'sec_app_open_acc': 2152680,
'sec_app_revol_util': 2154517,
'sec_app_open_act_il': 2152680,
'sec_app_num_rev_accts': 2152680,
'sec_app_chargeoff_within_12_mths': 2152680,
'sec_app_collections_12_mths_ex_med': 2152680,
'sec_app_mths_since_last_major_derog': 2224759,
'hardship_flag': 33,
'hardship_type': 2249784,
'hardship_reason': 2249784,
'hardship_status': 2249784,
'deferral_term': 2249784,
'hardship_amount': 2249784,
'hardship_start_date': 2249784,
'hardship_end_date': 2249784,
'payment_plan_start_date': 2249784,
'hardship_length': 2249784,
'hardship_dpd': 2249784,
'hardship_loan_status': 2249784,
'orig_projected_additional_accrued_interest': 2252050,
'hardship_payoff_balance_amount': 2249784,
'hardship_last_payment_amount': 2249784,
'disbursement_method': 33,
'debt_settlement_flag': 33,
'debt_settlement_flag_date': 2226455,
'settlement_status': 2226455,
'settlement_date': 2226455,
'settlement_amount': 2226455,
'settlement_percentage': 2226455,
'settlement_term': 2226455}
```

3 to get all the columns missing values percentage at single instance

[21]:
```
#to get all the columns missing values percentage at single instance
def check_missing_value_pcntg(data,lst_cl):
    missing_values_less_than_75 = {}
    missing_values_greater_than_75 = {}
    b = data.count()
    for i in lst_cl:
        a = df.filter(col(i).isNull()).count()
        c = (a/b) * 100
        if c >= 75:
            missing_values_greater_than_75[i] = c
        else:
            missing_values_less_than_75[i] = c
    return ({"Missing_Value_above75:":
 ↪missing_values_greater_than_75,"Missing_Value_below75:":
 ↪missing_values_less_than_75})
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 23, Finished, Available,␣
 ↪Finished)
```

[22]:
```
check_missing_value_pcntg(df,df.columns) #checking the missing values␣
 ↪percentage of each columns
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 24, Finished, Available,␣
 ↪Finished)
```

[22]:
```
{'Missing_Value_above75:': {'member_id': 100.0,
  'desc': 94.42354384768265,
  'mths_since_last_record': 84.11306935326698,
  'annual_inc_joint': 94.66050574578416,
  'dti_joint': 94.66068268205305,
  'verification_status_joint': 94.88079140054346,
  'mths_since_recent_bc_dlq': 77.01151103131285,
  'revol_bal_joint': 95.22183605881538,
  'sec_app_fico_range_low': 95.22179182474817,
  'sec_app_fico_range_high': 95.22179182474817,
  'sec_app_earliest_cr_line': 95.22179182474817,
  'sec_app_inq_last_6mths': 95.22179182474817,
  'sec_app_mort_acc': 95.22179182474817,
  'sec_app_open_acc': 95.22179182474817,
  'sec_app_revol_util': 95.30304980623266,
  'sec_app_open_act_il': 95.22179182474817,
  'sec_app_num_rev_accts': 95.22179182474817,
  'sec_app_chargeoff_within_12_mths': 95.22179182474817,
  'sec_app_collections_12_mths_ex_med': 95.22179182474817,
```

```
 'sec_app_mths_since_last_major_derog': 98.41013915595207,
 'hardship_type': 99.51709668815116,
 'hardship_reason': 99.51709668815116,
 'hardship_status': 99.51709668815116,
 'deferral_term': 99.51709668815116,
 'hardship_amount': 99.51709668815116,
 'hardship_start_date': 99.51709668815116,
 'hardship_end_date': 99.51709668815116,
 'payment_plan_start_date': 99.51709668815116,
 'hardship_length': 99.51709668815116,
 'hardship_dpd': 99.51709668815116,
 'hardship_loan_status': 99.51709668815116,
 'orig_projected_additional_accrued_interest': 99.61733108447336,
 'hardship_payoff_balance_amount': 99.51709668815116,
 'hardship_last_payment_amount': 99.51709668815116,
 'debt_settlement_flag_date': 98.48516013395844,
 'settlement_status': 98.48516013395844,
 'settlement_date': 98.48516013395844,
 'settlement_amount': 98.48516013395844,
 'settlement_percentage': 98.48516013395844,
 'settlement_term': 98.48516013395844},
'Missing_Value_below75:': {'id': 0.0,
 'loan_amnt': 0.0014597242182845054,
 'funded_amnt': 0.0014597242182845054,
 'funded_amnt_inv': 0.0014597242182845054,
 'term': 0.0014597242182845054,
 'int_rate': 0.0014597242182845054,
 'installment': 0.0014597242182845054,
 'grade': 0.0014597242182845054,
 'sub_grade': 0.0014597242182845054,
 'emp_title': 7.387177693998455,
 'emp_length': 6.499753837415917,
 'home_ownership': 0.0014597242182845054,
 'annual_inc': 0.001636660487167476,
 'verification_status': 0.0014597242182845054,
 'issue_d': 0.0014597242182845054,
 'loan_status': 0.0014597242182845054,
 'pymnt_plan': 0.0014597242182845054,
 'url': 0.0014597242182845054,
 'purpose': 0.0014597242182845054,
 'title': 1.0332193421421054,
 'zip_code': 0.0015039582855052483,
 'addr_state': 0.0014597242182845054,
 'dti': 0.07714421323297507,
 'delinq_2yrs': 0.0027425121676860407,
 'earliest_cr_line': 0.0027425121676860407,
 'fico_range_low': 0.0014597242182845054,
```

```
'fico_range_high': 0.0014597242182845054,
'inq_last_6mths': 0.0027867462349067834,
'mths_since_last_delinq': 51.24671506758302,
'open_acc': 0.0027425121676860407,
'pub_rec': 0.0027425121676860407,
'revol_bal': 0.0014597242182845054,
'revol_util': 0.08116951335006266,
'total_acc': 0.0027425121676860407,
'initial_list_status': 0.0014597242182845054,
'out_prncp': 0.0014597242182845054,
'out_prncp_inv': 0.0014597242182845054,
'total_pymnt': 0.0014597242182845054,
'total_pymnt_inv': 0.0014597242182845054,
'total_rec_prncp': 0.0014597242182845054,
'total_rec_int': 0.0014597242182845054,
'total_rec_late_fee': 0.0014597242182845054,
'recoveries': 0.0014597242182845054,
'collection_recovery_fee': 0.0014597242182845054,
'last_pymnt_d': 0.10881580536302679,
'last_pymnt_amnt': 0.0014597242182845054,
'next_pymnt_d': 59.50999269695551,
'last_credit_pull_d': 0.004644577058177972,
'last_fico_range_high': 0.0014597242182845054,
'last_fico_range_low': 0.0014597242182845054,
'collections_12_mths_ex_med': 0.007873663965292182,
'mths_since_last_major_derog': 74.30995960987322,
'policy_code': 0.0014597242182845054,
'application_type': 0.0014597242182845054,
'acc_now_delinq': 0.0027425121676860407,
'tot_coll_amt': 3.110053032223191,
'tot_cur_bal': 3.110053032223191,
'open_acc_6m': 38.313912366120064,
'open_act_il': 38.313868132052846,
'open_il_12m': 38.313868132052846,
'open_il_24m': 38.313868132052846,
'mths_since_rcnt_il': 40.25109910598527,
'total_bal_il': 38.313868132052846,
'il_util': 47.281042473109004,
'open_rv_12m': 38.313868132052846,
'open_rv_24m': 38.313868132052846,
'max_bal_bc': 38.313868132052846,
'all_util': 38.32355539277419,
'total_rev_hi_lim': 3.110053032223191,
'inq_fi': 38.313868132052846,
'total_cu_tl': 38.313912366120064,
'inq_last_12m': 38.313912366120064,
'acc_open_past_24mths': 2.214490107272036,
```

```
 'avg_cur_bal': 3.1131494169286427,
 'bc_open_to_buy': 3.316139551404631,
 'bc_util': 3.3663894517673945,
 'chargeoff_within_12_mths': 0.007873663965292182,
 'delinq_amnt': 0.0027425121676860407,
 'mo_sin_old_il_acct': 6.153135686674178,
 'mo_sin_old_rev_tl_op': 3.1100972662904116,
 'mo_sin_rcnt_rev_tl_op': 3.1100972662904116,
 'mo_sin_rcnt_tl': 3.110053032223191,
 'mort_acc': 2.214490107272036,
 'mths_since_recent_bc': 3.2487710670274392,
 'mths_since_recent_inq': 13.069751373578372,
 'mths_since_recent_revol_delinq': 67.25091022651823,
 'num_accts_ever_120_pd': 3.110053032223191,
 'num_actv_bc_tl': 3.110053032223191,
 'num_actv_rev_tl': 3.110053032223191,
 'num_bc_sats': 2.593133722681593,
 'num_bc_tl': 3.110053032223191,
 'num_il_tl': 3.110053032223191,
 'num_op_rev_tl': 3.110053032223191,
 'num_rev_accts': 3.1100972662904116,
 'num_rev_tl_bal_gt_0': 3.110053032223191,
 'num_sats': 2.593133722681593,
 'num_tl_120dpd_2m': 6.798333791155929,
 'num_tl_30dpd': 3.110053032223191,
 'num_tl_90g_dpd_24m': 3.110053032223191,
 'num_tl_op_past_12m': 3.110053032223191,
 'pct_tl_nvr_dlq': 3.116909312642406,
 'percent_bc_gt_75': 3.3357794772506404,
 'pub_rec_bankruptcies': 0.061839225974598136,
 'tax_liens': 0.006104301276462478,
 'tot_hi_cred_lim': 3.110053032223191,
 'total_bal_ex_mort': 2.214490107272036,
 'total_bc_limit': 2.214490107272036,
 'total_il_high_credit_limit': 3.110053032223191,
 'hardship_flag': 0.0014597242182845054,
 'disbursement_method': 0.0014597242182845054,
 'debt_settlement_flag': 0.0014597242182845054}}
```

[23]:
```
miss = check_missing_value_pcntg(df,df.columns) #assigning to df
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 25, Finished, Available,␣
 ↪Finished)
```

[24]:
```
column_names = list(miss['Missing_Value_above75:'].keys())
print("columnNames:",column_names)                              #extratcing␣
 ↪columns that nulls more than 75%
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 26, Finished, Available,␣
↪Finished)
```

```
columnNames: ['member_id', 'desc', 'mths_since_last_record', 'annual_inc_joint',
'dti_joint', 'verification_status_joint', 'mths_since_recent_bc_dlq',
'revol_bal_joint', 'sec_app_fico_range_low', 'sec_app_fico_range_high',
'sec_app_earliest_cr_line', 'sec_app_inq_last_6mths', 'sec_app_mort_acc',
'sec_app_open_acc', 'sec_app_revol_util', 'sec_app_open_act_il',
'sec_app_num_rev_accts', 'sec_app_chargeoff_within_12_mths',
'sec_app_collections_12_mths_ex_med', 'sec_app_mths_since_last_major_derog',
'hardship_type', 'hardship_reason', 'hardship_status', 'deferral_term',
'hardship_amount', 'hardship_start_date', 'hardship_end_date',
'payment_plan_start_date', 'hardship_length', 'hardship_dpd',
'hardship_loan_status', 'orig_projected_additional_accrued_interest',
'hardship_payoff_balance_amount', 'hardship_last_payment_amount',
'debt_settlement_flag_date', 'settlement_status', 'settlement_date',
'settlement_amount', 'settlement_percentage', 'settlement_term']
```

4 cols_nulls_morethan_75pct

[25]:
```
cols_nulls_morethan_75pct = ['member_id', 'desc', 'mths_since_last_record',␣
↪'annual_inc_joint', 'dti_joint', 'verification_status_joint',␣
↪'mths_since_recent_bc_dlq', 'revol_bal_joint', 'sec_app_fico_range_low',␣
↪'sec_app_fico_range_high', 'sec_app_earliest_cr_line',␣
↪'sec_app_inq_last_6mths', 'sec_app_mort_acc', 'sec_app_open_acc',␣
↪'sec_app_revol_util', 'sec_app_open_act_il', 'sec_app_num_rev_accts',␣
↪'sec_app_chargeoff_within_12_mths', 'sec_app_collections_12_mths_ex_med',␣
↪'sec_app_mths_since_last_major_derog', 'hardship_type', 'hardship_reason',␣
↪'hardship_status', 'deferral_term', 'hardship_amount',␣
↪'hardship_start_date', 'hardship_end_date', 'payment_plan_start_date',␣
↪'hardship_length', 'hardship_dpd', 'hardship_loan_status',␣
↪'orig_projected_additional_accrued_interest',␣
↪'hardship_payoff_balance_amount', 'hardship_last_payment_amount',␣
↪'debt_settlement_flag_date', 'settlement_status', 'settlement_date',␣
↪'settlement_amount', 'settlement_percentage', 'settlement_term']
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 27, Finished, Available,␣
↪Finished)
```

5 To chec duplicates

[26]:
```
def check_dups(data,cl):
    a = data.select(cl).distinct().count()
    b = data.count()
    if a == b:
        print("No Duplicates")
    else:
```

```
        c = b - a
        print("There are", c, "Duplicates")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 28, Finished, Available,␣
↪Finished)
```

[27]:
```
check_dups(df,"id")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 29, Finished, Available,␣
↪Finished)
```

```
No Duplicates
```

[28]:
```
check_dups(df,"num_bc_tl")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 30, Finished, Available,␣
↪Finished)
```

```
There are 2260624 Duplicates
```

6 To check all column Duplicates at single instance

[29]:
```
def duplicate_check_allclms(data):
    for cl in df.columns:                                #writing fro loop
        a = data.select(cl).distinct().count()     #define variable
        b = data.count()                                 #define variable
        if a == b:
            print(f"No Duplicates in column: {cl}")
        else:
            c = b-a
            print(f"There are {c} duplicates in column: {cl}")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 31, Finished, Available,␣
↪Finished)
```

[30]:
```
duplicate_check_allclms(df)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 32, Finished, Available,␣
↪Finished)
```

```
No Duplicates in column: id
There are 2260700 duplicates in column: member_id
There are 2259128 duplicates in column: loan_amnt
There are 2259128 duplicates in column: funded_amnt
There are 2250643 duplicates in column: funded_amnt_inv
There are 2260698 duplicates in column: term
There are 2260027 duplicates in column: int_rate
There are 2167399 duplicates in column: installment
There are 2260693 duplicates in column: grade
```

```
There are 2260665 duplicates in column: sub_grade
There are 1748006 duplicates in column: emp_title
There are 2260689 duplicates in column: emp_length
There are 2260694 duplicates in column: home_ownership
There are 2171332 duplicates in column: annual_inc
There are 2260697 duplicates in column: verification_status
There are 2260561 duplicates in column: issue_d
There are 2260691 duplicates in column: loan_status
There are 2260698 duplicates in column: pymnt_plan
There are 32 duplicates in column: url
There are 2136199 duplicates in column: desc
There are 2260686 duplicates in column: purpose
There are 2197545 duplicates in column: title
There are 2259744 duplicates in column: zip_code
There are 2260649 duplicates in column: addr_state
There are 2249855 duplicates in column: dti
There are 2260663 duplicates in column: delinq_2yrs
There are 2259946 duplicates in column: earliest_cr_line
There are 2260652 duplicates in column: fico_range_low
There are 2260652 duplicates in column: fico_range_high
There are 2260672 duplicates in column: inq_last_6mths
There are 2260527 duplicates in column: mths_since_last_delinq
There are 2260571 duplicates in column: mths_since_last_record
There are 2260609 duplicates in column: open_acc
There are 2260657 duplicates in column: pub_rec
There are 2158449 duplicates in column: revol_bal
There are 2259270 duplicates in column: revol_util
There are 2260548 duplicates in column: total_acc
There are 2260698 duplicates in column: initial_list_status
There are 1904559 duplicates in column: out_prncp
There are 1892219 duplicates in column: out_prncp_inv
There are 626822 duplicates in column: total_pymnt
There are 949601 duplicates in column: total_pymnt_inv
There are 1774237 duplicates in column: total_rec_prncp
There are 1624779 duplicates in column: total_rec_int
There are 2242325 duplicates in column: total_rec_late_fee
There are 2127923 duplicates in column: recoveries
There are 2114478 duplicates in column: collection_recovery_fee
There are 2260564 duplicates in column: last_pymnt_d
There are 1556233 duplicates in column: last_pymnt_amnt
There are 2260594 duplicates in column: next_pymnt_d
There are 2260559 duplicates in column: last_credit_pull_d
There are 2260628 duplicates in column: last_fico_range_high
There are 2260629 duplicates in column: last_fico_range_low
There are 2260684 duplicates in column: collections_12_mths_ex_med
There are 2260517 duplicates in column: mths_since_last_major_derog
There are 2260699 duplicates in column: policy_code
There are 2260698 duplicates in column: application_type
```

```
There are 2243067 duplicates in column: annual_inc_joint
There are 2256682 duplicates in column: dti_joint
There are 2260697 duplicates in column: verification_status_joint
There are 2260691 duplicates in column: acc_now_delinq
There are 2245126 duplicates in column: tot_coll_amt
There are 1773012 duplicates in column: tot_cur_bal
There are 2260681 duplicates in column: open_acc_6m
There are 2260646 duplicates in column: open_act_il
There are 2260681 duplicates in column: open_il_12m
There are 2260669 duplicates in column: open_il_24m
There are 2260295 duplicates in column: mths_since_rcnt_il
There are 2098451 duplicates in column: total_bal_il
There are 2260420 duplicates in column: il_util
There are 2260671 duplicates in column: open_rv_12m
There are 2260650 duplicates in column: open_rv_24m
There are 2226974 duplicates in column: max_bal_bc
There are 2260512 duplicates in column: all_util
There are 2226480 duplicates in column: total_rev_hi_lim
There are 2260667 duplicates in column: inq_fi
There are 2260638 duplicates in column: total_cu_tl
There are 2260652 duplicates in column: inq_last_12m
There are 2260643 duplicates in column: acc_open_past_24mths
There are 2172103 duplicates in column: avg_cur_bal
There are 2169200 duplicates in column: bc_open_to_buy
There are 2259206 duplicates in column: bc_util
There are 2260689 duplicates in column: chargeoff_within_12_mths
There are 2258083 duplicates in column: delinq_amnt
There are 2260134 duplicates in column: mo_sin_old_il_acct
There are 2259913 duplicates in column: mo_sin_old_rev_tl_op
There are 2260367 duplicates in column: mo_sin_rcnt_rev_tl_op
There are 2260468 duplicates in column: mo_sin_rcnt_tl
There are 2260653 duplicates in column: mort_acc
There are 2260154 duplicates in column: mths_since_recent_bc
There are 2260523 duplicates in column: mths_since_recent_bc_dlq
There are 2260674 duplicates in column: mths_since_recent_inq
There are 2260521 duplicates in column: mths_since_recent_revol_delinq
There are 2260656 duplicates in column: num_accts_ever_120_pd
There are 2260658 duplicates in column: num_actv_bc_tl
There are 2260643 duplicates in column: num_actv_rev_tl
There are 2260640 duplicates in column: num_bc_sats
There are 2260624 duplicates in column: num_bc_tl
There are 2260578 duplicates in column: num_il_tl
There are 2260619 duplicates in column: num_op_rev_tl
There are 2260583 duplicates in column: num_rev_accts
There are 2260650 duplicates in column: num_rev_tl_bal_gt_0
There are 2260609 duplicates in column: num_sats
There are 2260693 duplicates in column: num_tl_120dpd_2m
There are 2260695 duplicates in column: num_tl_30dpd
```

```
There are 2260666 duplicates in column: num_tl_90g_dpd_24m
There are 2260667 duplicates in column: num_tl_op_past_12m
There are 2260010 duplicates in column: pct_tl_nvr_dlq
There are 2260416 duplicates in column: percent_bc_gt_75
There are 2260688 duplicates in column: pub_rec_bankruptcies
There are 2260658 duplicates in column: tax_liens
There are 1730728 duplicates in column: tot_hi_cred_lim
There are 2047923 duplicates in column: total_bal_ex_mort
There are 2240391 duplicates in column: total_bc_limit
There are 2066563 duplicates in column: total_il_high_credit_limit
There are 2203825 duplicates in column: revol_bal_joint
There are 2260638 duplicates in column: sec_app_fico_range_low
There are 2260638 duplicates in column: sec_app_fico_range_high
There are 2260037 duplicates in column: sec_app_earliest_cr_line
There are 2260693 duplicates in column: sec_app_inq_last_6mths
There are 2260677 duplicates in column: sec_app_mort_acc
There are 2260633 duplicates in column: sec_app_open_acc
There are 2259484 duplicates in column: sec_app_revol_util
There are 2260660 duplicates in column: sec_app_open_act_il
There are 2260614 duplicates in column: sec_app_num_rev_accts
There are 2260678 duplicates in column: sec_app_chargeoff_within_12_mths
There are 2260682 duplicates in column: sec_app_collections_12_mths_ex_med
There are 2260560 duplicates in column: sec_app_mths_since_last_major_derog
There are 2260698 duplicates in column: hardship_flag
There are 2260699 duplicates in column: hardship_type
There are 2260691 duplicates in column: hardship_reason
There are 2260697 duplicates in column: hardship_status
There are 2260699 duplicates in column: deferral_term
There are 2251538 duplicates in column: hardship_amount
There are 2260673 duplicates in column: hardship_start_date
There are 2260672 duplicates in column: hardship_end_date
There are 2260673 duplicates in column: payment_plan_start_date
There are 2260699 duplicates in column: hardship_length
There are 2260666 duplicates in column: hardship_dpd
There are 2260695 duplicates in column: hardship_loan_status
There are 2253213 duplicates in column:
orig_projected_additional_accrued_interest
There are 2249807 duplicates in column: hardship_payoff_balance_amount
There are 2251655 duplicates in column: hardship_last_payment_amount
There are 2260698 duplicates in column: disbursement_method
There are 2260698 duplicates in column: debt_settlement_flag
There are 2260617 duplicates in column: debt_settlement_flag_date
There are 2260697 duplicates in column: settlement_status
There are 2260610 duplicates in column: settlement_date
There are 2238759 duplicates in column: settlement_amount
There are 2258630 duplicates in column: settlement_percentage
There are 2260660 duplicates in column: settlement_term
```

[31]:
```
df.select("id").distinct().orderBy(asc('id')).show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 33, Finished, Available,␣
↪Finished)
```

```
+---------+
|       id|
+---------+
|  1000007|
|100001133|
|100001137|
|100001142|
|100001158|
+---------+
only showing top 5 rows
```

[32]:
```
df.select('id').distinct().orderBy(desc('id')).show(40,False)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 34, Finished, Available,␣
↪Finished)
```

```
+-----------------------------------------------+
|id                                             |
+-----------------------------------------------+
|Total amount funded in policy code 2: 873652739 |
|Total amount funded in policy code 2: 823319310 |
|Total amount funded in policy code 2: 820109297 |
|Total amount funded in policy code 2: 81866225  |
|Total amount funded in policy code 2: 737901574 |
|Total amount funded in policy code 2: 662815446 |
|Total amount funded in policy code 2: 651669342 |
|Total amount funded in policy code 2: 620899600 |
|Total amount funded in policy code 2: 608903141 |
|Total amount funded in policy code 2: 567447023 |
|Total amount funded in policy code 2: 564202131 |
|Total amount funded in policy code 2: 521953170 |
|Total amount funded in policy code 2: 520780182 |
|Total amount funded in policy code 2: 511988838 |
|Total amount funded in policy code 2: 1944088810|
|Total amount funded in policy code 2: 0         |
|Total amount funded in policy code 1: 6417608175|
|Total amount funded in policy code 1: 460296150 |
|Total amount funded in policy code 1: 3503840175|
|Total amount funded in policy code 1: 2700702175|
|Total amount funded in policy code 1: 2087217200|
|Total amount funded in policy code 1: 2080429200|
|Total amount funded in policy code 1: 2063142975|
```

```
|Total amount funded in policy code 1: 2050909275|
|Total amount funded in policy code 1: 1817354125|
|Total amount funded in policy code 1: 1791201400|
|Total amount funded in policy code 1: 1741781700|
|Total amount funded in policy code 1: 1538432075|
|Total amount funded in policy code 1: 1465324575|
|Total amount funded in policy code 1: 1443412975|
|Total amount funded in policy code 1: 1437969475|
|Total amount funded in policy code 1: 1404586950|
|Loans that do not meet the credit policy        |
|999989                                          |
|999983                                          |
|999981                                          |
|99997759                                        |
|99997746                                        |
|99997737                                        |
|99997727                                        |
+------------------------------------------------+
only showing top 40 rows
```

[33]:
```
df.select('id').filter(col('id').like("%Total amount funded in policy code%")).
  ↪count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 35, Finished, Available,␣
  ↪Finished)
```

[33]: 32

[34]:
```
display(df.filter(col('id').like("%Total amount funded in policy code%")))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 36, Finished, Available,␣
  ↪Finished)
```

```
SynapseWidget(Synapse.DataFrame, 0a049002-f7bb-40a0-b499-ae502b5c8977)
```

[35]:
```
df.select("*").filter(~col('id').rlike("^[A-Za-z]")).count() #to filter the␣
  ↪null rows from ID column having string values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 37, Finished, Available,␣
  ↪Finished)
```

[35]: 2260668

[36]:
```
df = df.filter(~col('id').rlike("^[A-Za-z]"))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 38, Finished, Available,␣
  ↪Finished)
```

[37]:
```
df.count() #33 rows removed having nulls  2260701 - 2260668
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 39, Finished, Available,␣
↪Finished)
```

[37]: 2260668

[38]:
```
df.select("term").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 40, Finished, Available,␣
↪Finished)
```

```
+----------+
|      term|
+----------+
| 36 months|
| 60 months|
+----------+
```

[39]:
```
df = df.withColumn("term_in_months", regexp_replace(col("term"), " months",""))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 41, Finished, Available,␣
↪Finished)
```

[40]:
```
df.select("term_in_months").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 42, Finished, Available,␣
↪Finished)
```

```
+--------------+
|term_in_months|
+--------------+
|            60|
|            36|
+--------------+
```

7 Columns to Drop

Term, emp_length,url,mths_since_last_major_derog

[41]:
```
#check the column have string datatype
for cl, dtype in df.dtypes:
    if dtype == 'string':
        print(f"Column '{cl}' has string values")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 43, Finished, Available,␣
↪Finished)
```

```
Column 'id' has string values
Column 'member_id' has string values
Column 'term' has string values
Column 'grade' has string values
Column 'sub_grade' has string values
Column 'emp_title' has string values
Column 'emp_length' has string values
Column 'home_ownership' has string values
Column 'verification_status' has string values
Column 'issue_d' has string values
Column 'loan_status' has string values
Column 'pymnt_plan' has string values
Column 'url' has string values
Column 'desc' has string values
Column 'purpose' has string values
Column 'title' has string values
Column 'zip_code' has string values
Column 'addr_state' has string values
Column 'earliest_cr_line' has string values
Column 'initial_list_status' has string values
Column 'last_pymnt_d' has string values
Column 'next_pymnt_d' has string values
Column 'last_credit_pull_d' has string values
Column 'application_type' has string values
Column 'verification_status_joint' has string values
Column 'sec_app_earliest_cr_line' has string values
Column 'hardship_flag' has string values
Column 'hardship_type' has string values
Column 'hardship_reason' has string values
Column 'hardship_status' has string values
Column 'hardship_start_date' has string values
Column 'hardship_end_date' has string values
Column 'payment_plan_start_date' has string values
Column 'hardship_loan_status' has string values
Column 'disbursement_method' has string values
Column 'debt_settlement_flag' has string values
Column 'debt_settlement_flag_date' has string values
Column 'settlement_status' has string values
Column 'settlement_date' has string values
Column 'term_in_months' has string values
```

[42]:
```
display(df.printSchema())
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 44, Finished, Available,␣
 ↪Finished)

root
 |-- id: string (nullable = true)
```

```
|-- member_id: string (nullable = true)
|-- loan_amnt: double (nullable = true)
|-- funded_amnt: double (nullable = true)
|-- funded_amnt_inv: double (nullable = true)
|-- term: string (nullable = true)
|-- int_rate: double (nullable = true)
|-- installment: double (nullable = true)
|-- grade: string (nullable = true)
|-- sub_grade: string (nullable = true)
|-- emp_title: string (nullable = true)
|-- emp_length: string (nullable = true)
|-- home_ownership: string (nullable = true)
|-- annual_inc: double (nullable = true)
|-- verification_status: string (nullable = true)
|-- issue_d: string (nullable = true)
|-- loan_status: string (nullable = true)
|-- pymnt_plan: string (nullable = true)
|-- url: string (nullable = true)
|-- desc: string (nullable = true)
|-- purpose: string (nullable = true)
|-- title: string (nullable = true)
|-- zip_code: string (nullable = true)
|-- addr_state: string (nullable = true)
|-- dti: double (nullable = true)
|-- delinq_2yrs: double (nullable = true)
|-- earliest_cr_line: string (nullable = true)
|-- fico_range_low: double (nullable = true)
|-- fico_range_high: double (nullable = true)
|-- inq_last_6mths: double (nullable = true)
|-- mths_since_last_delinq: double (nullable = true)
|-- mths_since_last_record: double (nullable = true)
|-- open_acc: double (nullable = true)
|-- pub_rec: double (nullable = true)
|-- revol_bal: double (nullable = true)
|-- revol_util: double (nullable = true)
|-- total_acc: double (nullable = true)
|-- initial_list_status: string (nullable = true)
|-- out_prncp: double (nullable = true)
|-- out_prncp_inv: double (nullable = true)
|-- total_pymnt: double (nullable = true)
|-- total_pymnt_inv: double (nullable = true)
|-- total_rec_prncp: double (nullable = true)
|-- total_rec_int: double (nullable = true)
|-- total_rec_late_fee: double (nullable = true)
|-- recoveries: double (nullable = true)
|-- collection_recovery_fee: double (nullable = true)
|-- last_pymnt_d: string (nullable = true)
|-- last_pymnt_amnt: double (nullable = true)
```

```
|-- next_pymnt_d: string (nullable = true)
|-- last_credit_pull_d: string (nullable = true)
|-- last_fico_range_high: double (nullable = true)
|-- last_fico_range_low: double (nullable = true)
|-- collections_12_mths_ex_med: double (nullable = true)
|-- mths_since_last_major_derog: double (nullable = true)
|-- policy_code: double (nullable = true)
|-- application_type: string (nullable = true)
|-- annual_inc_joint: double (nullable = true)
|-- dti_joint: double (nullable = true)
|-- verification_status_joint: string (nullable = true)
|-- acc_now_delinq: double (nullable = true)
|-- tot_coll_amt: double (nullable = true)
|-- tot_cur_bal: double (nullable = true)
|-- open_acc_6m: double (nullable = true)
|-- open_act_il: double (nullable = true)
|-- open_il_12m: double (nullable = true)
|-- open_il_24m: double (nullable = true)
|-- mths_since_rcnt_il: double (nullable = true)
|-- total_bal_il: double (nullable = true)
|-- il_util: double (nullable = true)
|-- open_rv_12m: double (nullable = true)
|-- open_rv_24m: double (nullable = true)
|-- max_bal_bc: double (nullable = true)
|-- all_util: double (nullable = true)
|-- total_rev_hi_lim: double (nullable = true)
|-- inq_fi: double (nullable = true)
|-- total_cu_tl: double (nullable = true)
|-- inq_last_12m: double (nullable = true)
|-- acc_open_past_24mths: double (nullable = true)
|-- avg_cur_bal: double (nullable = true)
|-- bc_open_to_buy: double (nullable = true)
|-- bc_util: double (nullable = true)
|-- chargeoff_within_12_mths: double (nullable = true)
|-- delinq_amnt: double (nullable = true)
|-- mo_sin_old_il_acct: double (nullable = true)
|-- mo_sin_old_rev_tl_op: double (nullable = true)
|-- mo_sin_rcnt_rev_tl_op: double (nullable = true)
|-- mo_sin_rcnt_tl: double (nullable = true)
|-- mort_acc: double (nullable = true)
|-- mths_since_recent_bc: double (nullable = true)
|-- mths_since_recent_bc_dlq: double (nullable = true)
|-- mths_since_recent_inq: double (nullable = true)
|-- mths_since_recent_revol_delinq: double (nullable = true)
|-- num_accts_ever_120_pd: double (nullable = true)
|-- num_actv_bc_tl: double (nullable = true)
|-- num_actv_rev_tl: double (nullable = true)
|-- num_bc_sats: double (nullable = true)
```

```
|-- num_bc_tl: double (nullable = true)
|-- num_il_tl: double (nullable = true)
|-- num_op_rev_tl: double (nullable = true)
|-- num_rev_accts: double (nullable = true)
|-- num_rev_tl_bal_gt_0: double (nullable = true)
|-- num_sats: double (nullable = true)
|-- num_tl_120dpd_2m: double (nullable = true)
|-- num_tl_30dpd: double (nullable = true)
|-- num_tl_90g_dpd_24m: double (nullable = true)
|-- num_tl_op_past_12m: double (nullable = true)
|-- pct_tl_nvr_dlq: double (nullable = true)
|-- percent_bc_gt_75: double (nullable = true)
|-- pub_rec_bankruptcies: double (nullable = true)
|-- tax_liens: double (nullable = true)
|-- tot_hi_cred_lim: double (nullable = true)
|-- total_bal_ex_mort: double (nullable = true)
|-- total_bc_limit: double (nullable = true)
|-- total_il_high_credit_limit: double (nullable = true)
|-- revol_bal_joint: double (nullable = true)
|-- sec_app_fico_range_low: double (nullable = true)
|-- sec_app_fico_range_high: double (nullable = true)
|-- sec_app_earliest_cr_line: string (nullable = true)
|-- sec_app_inq_last_6mths: double (nullable = true)
|-- sec_app_mort_acc: double (nullable = true)
|-- sec_app_open_acc: double (nullable = true)
|-- sec_app_revol_util: double (nullable = true)
|-- sec_app_open_act_il: double (nullable = true)
|-- sec_app_num_rev_accts: double (nullable = true)
|-- sec_app_chargeoff_within_12_mths: double (nullable = true)
|-- sec_app_collections_12_mths_ex_med: double (nullable = true)
|-- sec_app_mths_since_last_major_derog: double (nullable = true)
|-- hardship_flag: string (nullable = true)
|-- hardship_type: string (nullable = true)
|-- hardship_reason: string (nullable = true)
|-- hardship_status: string (nullable = true)
|-- deferral_term: double (nullable = true)
|-- hardship_amount: double (nullable = true)
|-- hardship_start_date: string (nullable = true)
|-- hardship_end_date: string (nullable = true)
|-- payment_plan_start_date: string (nullable = true)
|-- hardship_length: double (nullable = true)
|-- hardship_dpd: double (nullable = true)
|-- hardship_loan_status: string (nullable = true)
|-- orig_projected_additional_accrued_interest: double (nullable = true)
|-- hardship_payoff_balance_amount: double (nullable = true)
|-- hardship_last_payment_amount: double (nullable = true)
|-- disbursement_method: string (nullable = true)
|-- debt_settlement_flag: string (nullable = true)
```

```
 |-- debt_settlement_flag_date: string (nullable = true)
 |-- settlement_status: string (nullable = true)
 |-- settlement_date: string (nullable = true)
 |-- settlement_amount: double (nullable = true)
 |-- settlement_percentage: double (nullable = true)
 |-- settlement_term: double (nullable = true)
 |-- term_in_months: string (nullable = true)
```

[43]:
```
df.select('grade').distinct().show() #getting distinct values in grade column
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 45, Finished, Available,␣
 ↪Finished)
```

```
+-----+
|grade|
+-----+
|    F|
|    E|
|    B|
|    D|
|    C|
|    A|
|    G|
+-----+
```

[44]:
```
#to confirm if grade have only 1 character
df.select("grade").filter(length(col("grade")) >1).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 46, Finished, Available,␣
 ↪Finished)
```

```
+-----+
|grade|
+-----+
+-----+
```

[45]:
```
df.select("sub_grade").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 47, Finished, Available,␣
 ↪Finished)
```

```
+---------+
|sub_grade|
+---------+
|       D5|
|       F2|
```

```
|       B4|
|       A2|
|       E4|
|       B2|
|       C3|
|       D1|
|       C4|
|       F1|
|       D3|
|       F5|
|       G2|
|       B1|
|       B3|
|       E5|
|       C5|
|       G3|
|       A4|
|       F4|
+---------+
only showing top 20 rows
```

[46]:
```
df.select("sub_grade").filter(length(col("sub_grade"))>2).show() ##to confirm␣
↪if sub_grade have only 2 character
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 48, Finished, Available,␣
↪Finished)
```

```
+---------+
|sub_grade|
+---------+
+---------+
```

[47]:
```
df.select("sub_grade").filter(length(col("sub_grade"))<2).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 49, Finished, Available,␣
↪Finished)
```

```
+---------+
|sub_grade|
+---------+
+---------+
```

[48]:
```
df.select("emp_title").filter(col('emp_title').isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 50, Finished, Available,␣
↪Finished)
```

[48]: 166969

[49]:
```
df.select("emp_title").distinct().show(10,False)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 51, Finished, Available,␣
↪Finished)

+-------------------------+
|emp_title                |
+-------------------------+
|Systems Administrator II |
|Physician                |
|CSR                      |
|Nutrition                |
|tool room attendant      |
|office admin             |
|Front End Web Developer  |
|machinist                |
|SUPERINTENDENT           |
|Implementation Consultant|
+-------------------------+
only showing top 10 rows
```

[50]:
```
df.groupBy("emp_title").count().orderBy(desc("count")).show(20)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 52, Finished, Available,␣
↪Finished)

+------------------+------+
|         emp_title| count|
+------------------+------+
|              NULL|166969|
|           Teacher| 38824|
|           Manager| 34298|
|             Owner| 21977|
|  Registered Nurse| 15867|
|            Driver| 14753|
|                RN| 14737|
|        Supervisor| 14297|
|             Sales| 13050|
|   Project Manager| 10971|
|    Office Manager|  9772|
|   General Manager|  9251|
|          Director|  8934|
|             owner|  8507|
|         President|  7660|
|          Engineer|  7304|
```

```
|           manager|  7060|
|           teacher|  6692|
|Operations Manager|  6128|
|    Vice President|  5874|
+------------------+------+
only showing top 20 rows
```

[51]:
```
df = df.fillna("Others", subset = ["emp_title"]) #fill nulls with some other␣
 ↪values in a columns by using the subset
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 53, Finished, Available,␣
 ↪Finished)
```

[52]:
```
df.groupBy("emp_title").count().orderBy(desc("count")).show(20)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 54, Finished, Available,␣
 ↪Finished)
```

```
+------------------+------+
|         emp_title| count|
+------------------+------+
|            Others|166969|
|           Teacher| 38824|
|           Manager| 34298|
|             Owner| 21977|
|  Registered Nurse| 15867|
|            Driver| 14753|
|                RN| 14737|
|        Supervisor| 14297|
|             Sales| 13050|
|   Project Manager| 10971|
|    Office Manager|  9772|
|   General Manager|  9251|
|          Director|  8934|
|             owner|  8507|
|         President|  7660|
|          Engineer|  7304|
|           manager|  7060|
|           teacher|  6692|
|Operations Manager|  6128|
|    Vice President|  5874|
+------------------+------+
only showing top 20 rows
```

[53]:
```
display(df.select("*").filter(col("emp_title").isNull())) #.show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 55, Finished, Available,␣
↪Finished)
```

```
SynapseWidget(Synapse.DataFrame, 83c9405f-e89d-4e78-839f-0ad43e02aa47)
```

[54]:
```
df = df.withColumn("emp_title",initcap(col("emp_title"))) #to capitalize the␣
↪string vlaues in a column
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 56, Finished, Available,␣
↪Finished)
```

[55]:
```
df.groupBy("emp_title").count().orderBy(desc("count")).show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 57, Finished, Available,␣
↪Finished)
```

```
+----------------+------+
|       emp_title| count|
+----------------+------+
|          Others|166970|
|         Teacher| 46125|
|         Manager| 42822|
|           Owner| 31740|
|Registered Nurse| 21407|
|          Driver| 20786|
|      Supervisor| 18560|
|           Sales| 17647|
|              Rn| 16672|
|  Office Manager| 13163|
+----------------+------+
only showing top 10 rows
```

[56]:
```
df.select('emp_title').distinct().count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 58, Finished, Available,␣
↪Finished)
```

[56]:
```
438350
```

[57]:
```
df.select("emp_length").distinct().show(20,False)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 59, Finished, Available,␣
↪Finished)
```

```
+----------+
|emp_length|
+----------+
|9 years   |
|5 years   |
```

```
|1 year    |
|2 years   |
|7 years   |
|8 years   |
|4 years   |
|6 years   |
|3 years   |
|10+ years |
|< 1 year  |
|NULL      |
+----------+
```

[58]:
```
#display(df.select("*").filter(col("emp_length") == ' reactors')) #.show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 60, Finished, Available,␣
↪Finished)
```

[59]:
```
df.select("home_ownership").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 61, Finished, Available,␣
↪Finished)
```

```
+--------------+
|home_ownership|
+--------------+
|           OWN|
|          RENT|
|      MORTGAGE|
|           ANY|
|         OTHER|
|          NONE|
+--------------+
```

[60]:
```
df = df.withColumn("emp_length_years",regexp_extract(col("emp_length"),␣
↪"\\d+",0)) #to extract the digits from emp_lengthcolumn
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 62, Finished, Available,␣
↪Finished)
```

[61]:
```
df.select("emp_length","emp_length_years").distinct().show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 63, Finished, Available,␣
↪Finished)
```

```
+----------+----------------+
|emp_length|emp_length_years|
+----------+----------------+
```

```
|   6 years|                6|
|   3 years|                3|
| 10+ years|               10|
|   7 years|                7|
|  < 1 year|                1|
|   4 years|                4|
|   8 years|                8|
|   2 years|                2|
|   9 years|                9|
|    1 year|                1|
+----------+-----------------+
only showing top 10 rows
```

[62]:
```
df.select("*").filter(col("emp_length_years").isNull()).count() # teo get the␣
↪null values count in emp_length_years column
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 64, Finished, Available,␣
↪Finished)
```

[62]:
```
146907
```

[63]:
```
display(df.filter(col("emp_length_years").isNull()).head(5)) #.show(20)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 65, Finished, Available,␣
↪Finished)
```

```
SynapseWidget(Synapse.DataFrame, d7a6d30f-b003-4a71-9621-798da8e10c7f)
```

[64]:
```
df = df.fillna("NA", subset = ["emp_length_years"]) #replaced the null values␣
↪with NA
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 66, Finished, Available,␣
↪Finished)
```

[65]:
```
df.select("*").filter(col("emp_length_years").isNull()).count() # now there is␣
↪no null values
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 67, Finished, Available,␣
↪Finished)
```

[65]:
```
0
```

[66]:
```
df.groupBy(col("emp_length_years")).count().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 68, Finished, Available,␣
↪Finished)
```

```
+---------------+------+
|emp_length_years| count|
+---------------+------+
|               7| 92695|
|               3|180753|
|               8| 91914|
|              NA|146907|
|               5|139698|
|               6|102628|
|               9| 79395|
|               1|338391|
|              10|748005|
|               4|136605|
|               2|203677|
+---------------+------+
```

[67]:
```
df.select("home_ownership").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 69, Finished, Available,␣
↪Finished)
```

```
+--------------+
|home_ownership|
+--------------+
|           OWN|
|          RENT|
|      MORTGAGE|
|           ANY|
|         OTHER|
|          NONE|
+--------------+
```

[68]:
```
df.select("verification_status").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 70, Finished, Available,␣
↪Finished)
```

```
+-------------------+
|verification_status|
+-------------------+
|           Verified|
|    Source Verified|
|       Not Verified|
+-------------------+
```

[69]:
```
df.select("loan_status").distinct().show(10,False)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 71, Finished, Available,␣
 ↪Finished)
```

```
+-----------------------------------------------------+
|loan_status                                          |
+-----------------------------------------------------+
|Fully Paid                                           |
|Default                                              |
|In Grace Period                                      |
|Charged Off                                          |
|Late (31-120 days)                                   |
|Current                                              |
|Late (16-30 days)                                    |
|Does not meet the credit policy. Status:Fully Paid   |
|Does not meet the credit policy. Status:Charged Off  |
+-----------------------------------------------------+
```

[70]:
```
df.select("title").distinct().count() #show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 72, Finished, Available,␣
 ↪Finished)
```

[70]: 63156

[71]:
```
df.select("title").distinct().orderBy(desc(col("title"))).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 73, Finished, Available,␣
 ↪Finished)
```

```
+--------------------+
|               title|
+--------------------+
|î î MY FIRST CA…|
|        ~Summer Fun~|
|~Life Reorganizat…|
|               zxcvb|
|        zonball Loan|
|              zipcar|
|          zeusamoose|
|            zerodebt|
|       zero interest|
|           zero dept|
|           zero debt|
|zero credit card …|
|        zero balance|
|                zero|
```

```
|          zandercade|
|                zack|
|your rate is bett…|
|     your helping me|
|   youngest daughter|
|young woman with …|
+--------------------+
only showing top 20 rows
```

[72]:
```
df.select(col('loan_amnt').isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 74, Finished, Available,␣
↪Finished)
```

[72]: 2260668

[73]:
```
df=df.fillna(0, subset = "loan_amnt")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 75, Finished, Available,␣
↪Finished)
```

[74]:
```
df.select("loan_amnt").filter(col("loan_amnt").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 76, Finished, Available,␣
↪Finished)
```

[74]: 0

[75]:
```
df.select("loan_amnt").distinct().orderBy(desc(col("loan_amnt"))).show(50)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 77, Finished, Available,␣
↪Finished)
```

```
+---------+
|loan_amnt|
+---------+
|  40000.0|
|  39975.0|
|  39950.0|
|  39925.0|
|  39900.0|
|  39875.0|
|  39850.0|
|  39825.0|
|  39800.0|
|  39775.0|
|  39750.0|
|  39725.0|
```

```
|  39700.0|
|  39675.0|
|  39650.0|
|  39625.0|
|  39600.0|
|  39575.0|
|  39550.0|
|  39525.0|
|  39500.0|
|  39475.0|
|  39450.0|
|  39425.0|
|  39400.0|
|  39375.0|
|  39350.0|
|  39325.0|
|  39300.0|
|  39275.0|
|  39250.0|
|  39225.0|
|  39200.0|
|  39175.0|
|  39150.0|
|  39125.0|
|  39100.0|
|  39075.0|
|  39050.0|
|  39025.0|
|  39000.0|
|  38975.0|
|  38950.0|
|  38925.0|
|  38900.0|
|  38875.0|
|  38850.0|
|  38825.0|
|  38800.0|
|  38775.0|
+---------+
only showing top 50 rows
```

[76]:
```
df.groupBy("loan_amnt").count().orderBy(desc(col("loan_amnt"))).show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 78, Finished, Available,␣
  ↪Finished)

+---------+-----+
```

```
|loan_amnt|count|
+---------+-----+
|  40000.0|33368|
|  39975.0|   11|
|  39950.0|   10|
|  39925.0|   14|
|  39900.0|   24|
+---------+-----+
only showing top 5 rows
```

[77]:
```
df = df.fillna(0, subset =␣
  ↪["funded_amnt","funded_amnt_inv","int_rate","installment","annual_inc"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 79, Finished, Available,␣
  ↪Finished)
```

[78]:
```
df.select("verification_status").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 80, Finished, Available,␣
  ↪Finished)
```

```
+-------------------+
|verification_status|
+-------------------+
|           Verified|
|    Source Verified|
|       Not Verified|
+-------------------+
```

[79]:
```
df.select("issue_d").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 81, Finished, Available,␣
  ↪Finished)
```

```
+--------+
| issue_d|
+--------+
|Oct-2016|
|Sep-2017|
|May-2015|
|Dec-2014|
|Mar-2018|
|Sep-2018|
|Jul-2015|
|Feb-2014|
|Sep-2015|
|Jan-2016|
```

```
|Nov-2017|
|Jan-2014|
|Jul-2018|
|Oct-2015|
|May-2016|
|Jan-2018|
|May-2014|
|Aug-2018|
|Apr-2014|
|Apr-2016|
+--------+
only showing top 20 rows
```

[80]:
```
df.select("loan_status").distinct().show(10,False)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 82, Finished, Available,␣
↪Finished)

+-----------------------------------------------------+
|loan_status                                          |
+-----------------------------------------------------+
|Fully Paid                                           |
|Default                                              |
|In Grace Period                                      |
|Charged Off                                          |
|Late (31-120 days)                                   |
|Current                                              |
|Late (16-30 days)                                    |
|Does not meet the credit policy. Status:Fully Paid   |
|Does not meet the credit policy. Status:Charged Off  |
+-----------------------------------------------------+
```

[81]:
```
df.select("pymnt_plan").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 83, Finished, Available,␣
↪Finished)

+----------+
|pymnt_plan|
+----------+
|         n|
|         y|
+----------+
```

[82]:
```
df.select("purpose").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 84, Finished, Available,␣
↪Finished)
```

```
+------------------+
|           purpose|
+------------------+
|           wedding|
|             other|
|    small_business|
|debt_consolidation|
|       credit_card|
|            moving|
|          vacation|
|  renewable_energy|
|             house|
|               car|
|    major_purchase|
|           medical|
|  home_improvement|
|       educational|
+------------------+
```

[83]:
```
df.groupby("purpose").count().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 85, Finished, Available,␣
↪Finished)
```

```
+------------------+-------+
|           purpose|  count|
+------------------+-------+
|           wedding|   2355|
|             other| 139440|
|    small_business|  24689|
|debt_consolidation|1277877|
|       credit_card| 516971|
|            moving|  15403|
|          vacation|  15525|
|  renewable_energy|   1445|
|             house|  14136|
|               car|  24013|
|    major_purchase|  50445|
|           medical|  27488|
|  home_improvement| 150457|
|       educational|    424|
+------------------+-------+
```

[84]:
```
df.groupBy("title").count().show(50)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 86, Finished, Available,␣
↪Finished)
```

```
+--------------------+------+
|               title| count|
+--------------------+------+
|       Payoff Card's|     1|
|Davey's consolida…|     1|
|Credit Consolidation|   435|
|              Payoff|   666|
|    Home Improvement|  1773|
|                loan|   453|
|              floors|     2|
|             My Loan|   566|
|            Personal|  1185|
|  Credit Card Payoff|  1386|
|my debt consolida…|    17|
|  Debt Consolidation| 15763|
|                NULL| 23325|
|                Bill|     7|
|             My loan|   246|
|              School|    21|
|      Major purchase| 44840|
|credit card refin…|   492|
|          second try|     3|
|Credit Card Conso…|  2360|
| Bill consolidation |     9|
|           Debt Help|    58|
|    Debt elimination|    10|
|       Personal Loan|  2133|
|       credit cards |    18|
|Out of Debt With …|     1|
|              debt 1|     3|
|                refi|   107|
|         consolidate|   763|
|               Other|127714|
|       mrlmalsrl1944|     1|
|                LOAN|    78|
|           CLEAN UP |     1|
|Credit card refin…|469691|
|Credit consolidation|   104|
|    Wedding expenses|   183|
|    PAY CREDIT CARDS|     9|
|      consolidate me|     4|
|  pay of cridet card|     1|
|     Be Healthy 2014|     1|
```

```
|       Consolidation|  5385|
|    CC Consolidation|   410|
|         Home buying| 12714|
|            bill pay|    40|
|    Home improvement|137437|
|Debt Consolidatio…|    87|
|             Freedom|   803|
|Credit Card Refin…|   154|
|      Credit pay off|    23|
|    Debt Consideration|    13|
+--------------------+------+
only showing top 50 rows
```

[85]:
```
df.select("zip_code").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 87, Finished, Available,␣
↪Finished)
```

```
+--------+
|zip_code|
+--------+
|   471xx|
|   418xx|
|   957xx|
|   223xx|
|   230xx|
|   143xx|
|   751xx|
|   154xx|
|   371xx|
|   591xx|
|   183xx|
|   831xx|
|   756xx|
|   287xx|
|   179xx|
|   535xx|
|   216xx|
|   895xx|
|   625xx|
|   387xx|
+--------+
only showing top 20 rows
```

[86]:
```
df.select("zip_code").filter(length(col("zip_code")) > 5).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 88, Finished, Available,␣
↪Finished)

+--------+
|zip_code|
+--------+
+--------+
```

[87]:
```
df.select("zip_code").filter(length(col("zip_code")) <5).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 89, Finished, Available,␣
↪Finished)

+--------+
|zip_code|
+--------+
+--------+
```

[88]:
```
df.select("zip_code").filter(col("zip_code").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 90, Finished, Available,␣
↪Finished)
```

[88]:
```
1
```

[89]:
```
df = df.fillna(0, subset = "zip_code")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 91, Finished, Available,␣
↪Finished)
```

[90]:
```
df.select("addr_state").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 92, Finished, Available,␣
↪Finished)

+----------+
|addr_state|
+----------+
|        SC|
|        AZ|
|        LA|
|        MN|
|        NJ|
|        DC|
|        OR|
|        VA|
|        RI|
|        KY|
```

```
|        WY|
|        NH|
|        MI|
|        NV|
|        WI|
|        ID|
|        CA|
|        NE|
|        CT|
|        MT|
+----------+
only showing top 20 rows
```

[91]:
```
df.select("addr_state").filter(length(col("addr_state")) > 2).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 93, Finished, Available,␣
↪Finished)
```

```
+----------+
|addr_state|
+----------+
+----------+
```

[92]:
```
df.select("addr_state").filter(length(col("addr_state"))<2).show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 94, Finished, Available,␣
↪Finished)
```

```
+----------+
|addr_state|
+----------+
+----------+
```

[93]:
```
df.select("addr_state").filter(col("addr_state").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 95, Finished, Available,␣
↪Finished)
```

[93]:
```
0
```

[94]:
```
df.select("dti").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 96, Finished, Available,␣
↪Finished)
```

```
+-----+
|  dti|
```

```
+-----+
|19.98|
| 9.13|
|30.49|
| 14.9|
|17.52|
|17.56|
| 13.4|
| 2.86|
|23.04|
|37.81|
| 8.51|
| 3.26|
|17.95|
|35.17|
| 15.5|
|26.72|
| 26.7|
|12.32|
|41.89|
|38.61|
+-----+
only showing top 20 rows
```

[95]:
```
df.select("dti").filter(col("dti").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 97, Finished, Available,␣
↪Finished)
```

[95]: 1711

[96]:
```
df=df.fillna(0, subset = "dti")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 98, Finished, Available,␣
↪Finished)
```

[97]:
```
df.select("dti").filter(col("dti").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 99, Finished, Available,␣
↪Finished)
```

[97]: 0

[98]:
```
df.select("delinq_2yrs").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 100, Finished, Available,␣
↪Finished)
```

```
+-----------+
|delinq_2yrs|
+-----------+
|        8.0|
|        0.0|
|        7.0|
|       29.0|
|       35.0|
|       18.0|
|        1.0|
|        4.0|
|       11.0|
|       58.0|
|       21.0|
|       14.0|
|       22.0|
|        3.0|
|       19.0|
|       28.0|
|        2.0|
|       17.0|
|       27.0|
|       10.0|
+-----------+
only showing top 20 rows
```

[99]:
```
df.select("delinq_2yrs").filter(col("delinq_2yrs").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 101, Finished, Available,␣
↪Finished)
```

[99]: 29

[100]:
```
df=df.fillna(0, subset = "delinq_2yrs")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 102, Finished, Available,␣
↪Finished)
```

[101]:
```
df.select("delinq_2yrs").filter(col("delinq_2yrs").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 103, Finished, Available,␣
↪Finished)
```

[101]: 0

[102]:
```
df.select("earliest_cr_line").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 104, Finished, Available,␣
↪Finished)

+----------------+
|earliest_cr_line|
+----------------+
|        Jan-1999|
|        Jul-1996|
|        Nov-1978|
|        May-1977|
|        Mar-1999|
|        Jul-1989|
|        Mar-1960|
|        Sep-1987|
|        Jun-1979|
|        May-1973|
|        Apr-1988|
|        Oct-1975|
|        May-1993|
|        Sep-1998|
|        Jun-1985|
|        Feb-1961|
|        Jan-1953|
|        Dec-1981|
|        Jun-1989|
|        Oct-1965|
+----------------+
only showing top 20 rows
```

[103]:
```
df.select("earliest_cr_line").filter(col("earliest_cr_line").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 105, Finished, Available,␣
↪Finished)
```

[103]: 29

[104]:
```
df=df.fillna("NA",subset = "earliest_cr_line")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 106, Finished, Available,␣
↪Finished)
```

[105]:
```
df.select("earliest_cr_line").filter(col("earliest_cr_line").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 107, Finished, Available,␣
↪Finished)
```

[105]: 0

[106]:
```
df.select("fico_range_low").filter(col("fico_range_low").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 108, Finished, Available,␣
↪Finished)
```

[106]: 0

[107]:
```
df.select("fico_range_high").filter(col("fico_range_high").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 109, Finished, Available,␣
↪Finished)
```

[107]: 0

[108]:
```
df.select("inq_last_6mths").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 110, Finished, Available,␣
↪Finished)
```

```
+--------------+
|inq_last_6mths|
+--------------+
|           0.0|
|           1.0|
|          NULL|
|           4.0|
|           3.0|
|           2.0|
|           6.0|
|           5.0|
|           8.0|
|           7.0|
|          18.0|
|          25.0|
|          31.0|
|          11.0|
|          14.0|
|          19.0|
|          28.0|
|          17.0|
|          27.0|
|          10.0|
+--------------+
only showing top 20 rows
```

[109]:
```
df.select("inq_last_6mths").filter(col("inq_last_6mths").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 111, Finished, Available,␣
↪Finished)
```

[109]:
```
30
```

[110]:
```
df = df.fillna(0, subset = "inq_last_6mths")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 112, Finished, Available,␣
↪Finished)
```

[111]:
```
df.select("inq_last_6mths").filter(col("inq_last_6mths").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 113, Finished, Available,␣
↪Finished)
```

[111]:
```
0
```

[112]:
```
df.select("mths_since_last_delinq").filter(col("mths_since_last_delinq").
↪isNull()).count() #
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 114, Finished, Available,␣
↪Finished)
```

[112]:
```
1158502
```

[113]:
```
df.select("open_acc").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 115, Finished, Available,␣
↪Finished)
```

```
+--------+
|open_acc|
+--------+
|     8.0|
|     0.0|
|     7.0|
|    49.0|
|    29.0|
|    64.0|
|    47.0|
|    42.0|
|    44.0|
|    35.0|
|    62.0|
|    18.0|
|     1.0|
|    39.0|
|    34.0|
|    37.0|
```

```
|    25.0|
|    36.0|
|    41.0|
|     4.0|
+--------+
only showing top 20 rows
```

[114]:
```
df.select("open_acc").filter(col("open_acc").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 116, Finished, Available,␣
↪Finished)
```

[114]: 29

[115]:
```
df.select("pub_rec").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 117, Finished, Available,␣
↪Finished)
```

```
+-------+
|pub_rec|
+-------+
|    8.0|
|    0.0|
|    7.0|
|   18.0|
|    1.0|
|   37.0|
|    4.0|
|   23.0|
|   11.0|
|   21.0|
|   14.0|
|   63.0|
|   22.0|
|    3.0|
|   19.0|
|    2.0|
|   17.0|
|   10.0|
|   40.0|
|   13.0|
+-------+
only showing top 20 rows
```

[116]:
```
df.select("pub_rec").filter(col("pub_rec").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 118, Finished, Available,␣
↪Finished)
```

[116]:
```
29
```

[117]:
```
df.select("revol_bal").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 119, Finished, Available,␣
↪Finished)
```

```
+---------+
|revol_bal|
+---------+
|  64193.0|
|   2862.0|
|    299.0|
|   5360.0|
|   4142.0|
|  28980.0|
|  17072.0|
|  39763.0|
|  14473.0|
|  32046.0|
|  20689.0|
|  21825.0|
|   6454.0|
|  16822.0|
|   8649.0|
|   7115.0|
|  10625.0|
|  11757.0|
|  11772.0|
|   1761.0|
+---------+
only showing top 20 rows
```

[118]:
```
df.select("revol_bal").filter(col("revol_bal").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 120, Finished, Available,␣
↪Finished)
```

[118]:
```
0
```

[119]:
```
df = df.fillna(0, subset = ["mths_since_last_delinq","pub_rec","open_acc"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 121, Finished, Available,␣
↪Finished)
```

[120]:
```
df.select("mths_since_last_delinq").filter(col("mths_since_last_delinq").
  ↪isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 122, Finished, Available,␣
  ↪Finished)
```

[120]: 0

[121]:
```
df.select("pub_rec").filter(col("pub_rec").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 123, Finished, Available,␣
  ↪Finished)
```

[121]: 0

[122]:
```
df.select("open_acc").filter(col("open_acc").isNull()).count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 124, Finished, Available,␣
  ↪Finished)
```

[122]: 0

[123]:
```
df.
  ↪select(["revol_util","initial_list_status","total_acc","out_prncp","out_prncp_inv"]).
  ↪show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 125, Finished, Available,␣
  ↪Finished)
```

```
+----------+------------------+---------+---------+-------------+
|revol_util|initial_list_status|total_acc|out_prncp|out_prncp_inv|
+----------+------------------+---------+---------+-------------+
|      72.0|                 w|     18.0|      0.0|          0.0|
|      58.0|                 f|     16.0|      0.0|          0.0|
|      84.0|                 w|     35.0|      0.0|          0.0|
|      59.0|                 w|     40.0|      0.0|          0.0|
|      87.0|                 w|     25.0|      0.0|          0.0|
|      89.0|                 w|     10.0|      0.0|          0.0|
|      95.0|                 w|     24.0|      0.0|          0.0|
|      94.0|                 w|     37.0|      0.0|          0.0|
|      95.0|                 w|     13.0|      0.0|          0.0|
|      57.0|                 w|     20.0|      0.0|          0.0|
+----------+------------------+---------+---------+-------------+
only showing top 10 rows
```

[124]:
```
df = df.fillna(0, subset =␣
  ↪["revol_util","total_acc","out_prncp","out_prncp_inv"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 126, Finished, Available,␣
 ↪Finished)
```

[125]:
```
df.select("initial_list_status").distinct().show(20)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 127, Finished, Available,␣
 ↪Finished)
```

```
+--------------------+
|initial_list_status|
+--------------------+
|                   f|
|                   w|
+--------------------+
```

[126]:
```
df.select("initial_list_status").filter(col("initial_list_status").isNull()).
 ↪count()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 128, Finished, Available,␣
 ↪Finished)
```

[126]:
```
0
```

[127]:
```
df.
 ↪select(["total_pymnt","total_pymnt_inv","total_rec_prncp","total_rec_int","total_rec_late_f
 ↪show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 129, Finished, Available,␣
 ↪Finished)
```

```
+-----------+---------------+---------------+-------------+------------------+
|total_pymnt|total_pymnt_inv|total_rec_prncp|total_rec_int|total_rec_late_fee|
+-----------+---------------+---------------+-------------+------------------+
|   14654.32|        14629.9|        6591.69|      5076.65|               0.0|
|    7021.05|        7021.05|        4370.75|      2022.29|               0.0|
|   10490.93|       10490.93|         4707.5|      3176.95|               0.0|
|   18783.18|       18783.18|        7127.22|      6564.21|               0.0|
|     4393.9|         4393.9|        2267.74|      1330.64|               0.0|
|     4422.4|         4422.4|        1336.56|       907.46|               0.0|
|   23745.99|       23745.99|       15765.81|      5535.75|               0.0|
|    4772.89|        4772.89|         794.07|      2028.97|               0.0|
|    2617.01|        2617.01|        1682.61|       638.18|               0.0|
|    8165.46|        8165.46|        3837.23|      1516.91|               0.0|
+-----------+---------------+---------------+-------------+------------------+
only showing top 10 rows
```

[128]:
```
df = df.
  ↪fillna(0,subset=["total_pymnt","total_pymnt_inv","total_rec_prncp","total_rec_int","total_r
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 130, Finished, Available,␣
  ↪Finished)
```

[130]:
```
df.
  ↪select(["recoveries","collection_recovery_fee","collection_recovery_fee","last_pymnt_d","la
  ↪show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 132, Finished, Available,␣
  ↪Finished)
```

```
+----------+----------------------+----------------------+------------+-------
--------+
|recoveries|collection_recovery_fee|collection_recovery_fee|last_pymnt_d|last_py
mnt_amnt|
+----------+----------------------+----------------------+------------+-------
--------+
|   2985.98|              537.4764|              537.4764|    Feb-2017|
687.68|
|    628.01|              113.0418|              113.0418|    Mar-2017|
356.48|
|   2606.48|              469.1664|              469.1664|    Aug-2016|
660.24|
|   5091.75|               916.515|               916.515|    Mar-2017|
723.34|
|    795.52|              143.1936|              143.1936|    May-2016|
360.83|
|   2178.38|              392.1084|              392.1084|    Dec-2015|
551.64|
|   2444.43|              439.9974|              439.9974|    Jan-2017|
1184.86|
|   1949.85|               350.973|               350.973|    Oct-2015|
972.74|
|    296.22|               53.3196|               53.3196|    Dec-2016|
136.7|
|   2811.32|              506.0376|              506.0376|    Feb-2016|
767.12|
+----------+----------------------+----------------------+------------+-------
--------+
only showing top 10 rows
```

[131]:
```
df = df.fillna(0, subset =␣
  ↪["recoveries","collection_recovery_fee","collection_recovery_fee","last_pymnt_amnt"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 133, Finished, Available,␣
↪Finished)
```

[132]:
```
df.select(["next_pymnt_d","last_credit_pull_d"]).show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 134, Finished, Available,␣
↪Finished)
```

```
+------------+------------------+
|next_pymnt_d|last_credit_pull_d|
+------------+------------------+
|        NULL|          Aug-2017|
|        NULL|          Jul-2018|
|        NULL|          Feb-2017|
|        NULL|          Aug-2017|
|        NULL|          Dec-2016|
|        NULL|          Oct-2016|
|        NULL|          Jul-2017|
|        NULL|          Oct-2017|
|        NULL|          Jun-2017|
|        NULL|          Oct-2016|
+------------+------------------+
only showing top 10 rows
```

[133]:
```
df =df.fillna("NA", subset =␣
↪["last_pymnt_d","next_pymnt_d","last_credit_pull_d"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 135, Finished, Available,␣
↪Finished)
```

[134]:
```
df.
↪select(["last_fico_range_high","last_fico_range_low","collections_12_mths_ex_med","mths_sin
↪show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 136, Finished, Available,␣
↪Finished)
```

```
+------------------+-----------------+-------------------------+-----------
---------------+
|last_fico_range_high|last_fico_range_low|collections_12_mths_ex_med|mths_since_
last_major_derog|
+------------------+-----------------+-------------------------+-----------
---------------+
|             504.0|            500.0|                      0.0|
NULL|
|             589.0|            585.0|                      0.0|
NULL|
|             574.0|            570.0|                      0.0|
```

```
NULL|
|                614.0|              610.0|                        0.0|
NULL|
|                534.0|              530.0|                        0.0|
NULL|
|                514.0|              510.0|                        0.0|
NULL|
|                554.0|              550.0|                        0.0|
NULL|
|                499.0|                0.0|                        0.0|
NULL|
|                564.0|              560.0|                        0.0|
NULL|
|                574.0|              570.0|                        0.0|
NULL|
+--------------------+--------------------+---------------------------+-----------
----------------+
only showing top 10 rows
```

[136]:
```
df.select("mths_since_last_major_derog").distinct().show(5) #need to dropoff
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 138, Finished, Available,␣
 ↪Finished)
```

```
+---------------------------+
|mths_since_last_major_derog|
+---------------------------+
|                      170.0|
|                      147.0|
|                      169.0|
|                      160.0|
|                       70.0|
+---------------------------+
only showing top 5 rows
```

[137]:
```
df=df.fillna(0,subset =␣
 ↪["last_fico_range_high","last_fico_range_low","collections_12_mths_ex_med"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 139, Finished, Available,␣
 ↪Finished)
```

[138]:
```
df.select(["policy_code","application_type","acc_now_delinq"]).show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 140, Finished, Available,␣
 ↪Finished)
```

```
+-----------+----------------+--------------+
```

```
|policy_code|application_type|acc_now_delinq|
+-----------+----------------+--------------+
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
|        1.0|      Individual|           0.0|
+-----------+----------------+--------------+
only showing top 10 rows
```

[139]:
```
df.select("application_type").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 141, Finished, Available,␣
 ↪Finished)
```

```
+----------------+
|application_type|
+----------------+
|       Joint App|
|      Individual|
+----------------+
```

[140]:
```
df=df.fillna(0, subset = ["policy_code","acc_now_delinq"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 142, Finished, Available,␣
 ↪Finished)
```

[141]:
```
df.select(["tot_cur_bal","tot_cur_bal"]).show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 143, Finished, Available,␣
 ↪Finished)
```

```
+-----------+-----------+
|tot_cur_bal|tot_cur_bal|
+-----------+-----------+
|   194764.0|   194764.0|
|    86501.0|    86501.0|
|   233225.0|   233225.0|
|   612285.0|   612285.0|
|   401020.0|   401020.0|
+-----------+-----------+
only showing top 5 rows
```

[142]:
```
df.select(["open_acc_6m","open_act_il","open_il_12m","open_il_24m"]).show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 144, Finished, Available,␣
↪Finished)
```

```
+-----------+-----------+-----------+-----------+
|open_acc_6m|open_act_il|open_il_12m|open_il_24m|
+-----------+-----------+-----------+-----------+
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
|       NULL|       NULL|       NULL|       NULL|
+-----------+-----------+-----------+-----------+
only showing top 10 rows
```

[143]:
```
df.select(["open_acc_6m","open_act_il","open_il_12m","open_il_24m"]).distinct().
↪show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 145, Finished, Available,␣
↪Finished)
```

```
+-----------+-----------+-----------+-----------+
|open_acc_6m|open_act_il|open_il_12m|open_il_24m|
+-----------+-----------+-----------+-----------+
|        4.0|        4.0|        4.0|        6.0|
|        2.0|        6.0|        1.0|        7.0|
|        2.0|        4.0|        4.0|        9.0|
|        3.0|        1.0|        4.0|        5.0|
|        3.0|       15.0|        0.0|        0.0|
|        2.0|        8.0|        2.0|        8.0|
|       10.0|        1.0|        1.0|        1.0|
|       11.0|        0.0|        0.0|        0.0|
|        3.0|        0.0|        1.0|        3.0|
|        0.0|       20.0|        3.0|        6.0|
+-----------+-----------+-----------+-----------+
only showing top 10 rows
```

[144]:
```
df = df.fillna(0, subset =␣
  ↪["open_acc_6m","open_act_il","open_il_12m","open_il_24m"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 146, Finished, Available,␣
  ↪Finished)
```

[146]:
```
df.
  ↪select(["mths_since_rcnt_il","total_bal_il","il_util","open_rv_12m","open_rv_24m","max_bal_
  ↪distinct().show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 148, Finished, Available,␣
  ↪Finished)
```

```
+------------------+------------+-------+-----------+-----------+----------+
|mths_since_rcnt_il|total_bal_il|il_util|open_rv_12m|open_rv_24m|max_bal_bc|
+------------------+------------+-------+-----------+-----------+----------+
|              16.0|     29135.0|   77.0|        3.0|        4.0|    1326.0|
|              19.0|     67955.0|  102.0|        2.0|        4.0|    3283.0|
|               1.0|     52907.0|   83.0|        3.0|        6.0|    3876.0|
|               8.0|     15629.0|   92.0|        2.0|        3.0|    2695.0|
|               3.0|     50367.0|   70.0|        0.0|        1.0|   10953.0|
|               8.0|     17919.0|   45.0|        1.0|        5.0|    2474.0|
|             125.0|         0.0|   NULL|        2.0|        3.0|    7540.0|
|              89.0|         0.0|   NULL|        4.0|        7.0|    5793.0|
|              13.0|     18517.0|   73.0|        0.0|        2.0|    4010.0|
|              17.0|     11923.0|   52.0|        1.0|        3.0|    3198.0|
+------------------+------------+-------+-----------+-----------+----------+
only showing top 10 rows
```

[147]:
```
df=df.fillna(0, subset =␣
  ↪["mths_since_rcnt_il","total_bal_il","il_util","open_rv_12m","open_rv_24m","max_bal_bc"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 149, Finished, Available,␣
  ↪Finished)
```

[148]:
```
df.
  ↪select(["all_util","total_rev_hi_lim","inq_fi","total_cu_tl","inq_last_12m","acc_open_past_
  ↪distinct().show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 150, Finished, Available,␣
  ↪Finished)
```

```
+--------+---------------+------+-----------+------------+--------------------+
|all_util|total_rev_hi_lim|inq_fi|total_cu_tl|inq_last_12m|acc_open_past_24mths|
+--------+---------------+------+-----------+------------+--------------------+
|    40.0|        47900.0|   1.0|        0.0|         3.0|                10.0|
|    51.0|        23700.0|   0.0|        0.0|         2.0|                 7.0|
|    NULL|       177900.0|  NULL|       NULL|        NULL|                 1.0|
```

```
|    NULL|         33700.0|  NULL|       NULL|        NULL|                 5.0|
|    NULL|         30000.0|  NULL|       NULL|        NULL|                 2.0|
|    NULL|         42500.0|  NULL|       NULL|        NULL|                 3.0|
|    NULL|         15100.0|  NULL|       NULL|        NULL|                 4.0|
|    NULL|          8500.0|  NULL|       NULL|        NULL|                 1.0|
|    NULL|         11300.0|  NULL|       NULL|        NULL|                 5.0|
|    NULL|         28750.0|  NULL|       NULL|        NULL|                 4.0|
+--------+----------------+------+-----------+------------+--------------------+
only showing top 10 rows
```

[149]:
```
df = df.fillna(0, subset =␣
  ↪["all_util","total_rev_hi_lim","inq_fi","total_cu_tl","inq_last_12m","acc_open_past_24mths"]
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 151, Finished, Available,␣
  ↪Finished)
```

[151]:
```
df.
  ↪select(["avg_cur_bal","bc_open_to_buy","bc_util","chargeoff_within_12_mths","delinq_amnt"])
  ↪distinct().show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 153, Finished, Available,␣
  ↪Finished)

+-----------+-------------+-------+-----------------------+-----------+
|avg_cur_bal|bc_open_to_buy|bc_util|chargeoff_within_12_mths|delinq_amnt|
+-----------+-------------+-------+-----------------------+-----------+
|    17629.0|        5742.0|   34.0|                    0.0|        0.0|
|     2806.0|        1943.0|   88.9|                    0.0|        0.0|
|     2268.0|         457.0|   86.9|                    0.0|        0.0|
|     4422.0|       32169.0|   35.9|                    0.0|        0.0|
|     8169.0|        4482.0|   75.8|                    0.0|        0.0|
|     5144.0|        4664.0|   86.6|                    0.0|        0.0|
|    30907.0|       11894.0|   63.8|                    0.0|        0.0|
|    13767.0|        4780.0|   75.5|                    0.0|        0.0|
|     3497.0|         482.0|   95.2|                    0.0|        0.0|
|     3746.0|        1528.0|   82.2|                    0.0|        0.0|
+-----------+-------------+-------+-----------------------+-----------+
only showing top 10 rows
```

[152]:
```
df =df.fillna(0,subset =␣
  ↪["avg_cur_bal","bc_open_to_buy","bc_util","chargeoff_within_12_mths","delinq_amnt"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 154, Finished, Available,␣
  ↪Finished)
```

[153]:
```
df.
  ↪select(["mo_sin_old_il_acct","mo_sin_old_rev_tl_op","mo_sin_rcnt_rev_tl_op","mo_sin_rcnt_tl
  ↪distinct().show(10)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 155, Finished, Available,␣
  ↪Finished)

+-----------------+------------------+--------------------+-------------+--
------+
|mo_sin_old_il_acct|mo_sin_old_rev_tl_op|mo_sin_rcnt_rev_tl_op|mo_sin_rcnt_tl|mo
rt_acc|
+-----------------+------------------+--------------------+-------------+--
------+
|             121.0|             157.0|                 9.0|          3.0|
2.0|
|              44.0|             310.0|                14.0|         11.0|
2.0|
|              45.0|              81.0|                 5.0|          5.0|
1.0|
|              80.0|             233.0|               141.0|         40.0|
3.0|
|             126.0|             107.0|                 8.0|          4.0|
0.0|
|              70.0|              82.0|                 8.0|          8.0|
0.0|
|              20.0|             205.0|                 0.0|          0.0|
0.0|
|              NULL|             194.0|                 4.0|          4.0|
0.0|
|             126.0|             277.0|                 1.0|          1.0|
1.0|
|             134.0|             113.0|                13.0|         12.0|
1.0|
+-----------------+------------------+--------------------+-------------+--
------+
only showing top 10 rows
```

[154]:
```
df=df.fillna(0, subset =␣
  ↪["mo_sin_old_il_acct","mo_sin_old_rev_tl_op","mo_sin_rcnt_rev_tl_op","mo_sin_rcnt_tl","mort
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 156, Finished, Available,␣
  ↪Finished)
```

[156]:
```
df.
  ↪select(["mths_since_recent_bc","mths_since_recent_inq","mths_since_recent_revol_delinq"]).
  ↪show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 158, Submitted, Running,␣
↪Running)
```

```
+--------------------+---------------------+-----------------------------+
|mths_since_recent_bc|mths_since_recent_inq|mths_since_recent_revol_delinq|
+--------------------+---------------------+-----------------------------+
|                 8.0|                  7.0|                         NULL|
|                21.0|                  1.0|                         NULL|
|                 4.0|                 19.0|                         NULL|
|                 2.0|                  2.0|                         NULL|
|                11.0|                  8.0|                         NULL|
+--------------------+---------------------+-----------------------------+
only showing top 5 rows
```

[157]:
```
df=df.fillna(0,subset =␣
↪["mths_since_recent_bc","mths_since_recent_inq","mths_since_recent_revol_delinq"])
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 159, Finished, Available,␣
↪Finished)
```

[158]:
```
df.
↪select(["num_accts_ever_120_pd","num_actv_bc_tl","num_actv_rev_tl","num_bc_sats","num_bc_tl
↪distinct().show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 160, Finished, Available,␣
↪Finished)
```

```
+--------------------+-------------+--------------+-----------+---------+----
-----+-------------+
|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|
+--------------------+-------------+--------------+-----------+---------+----
-----+-------------+
|                 0.0|          8.0|          12.0|        9.0|      9.0|
0.0|         13.0|
|                 0.0|          1.0|           4.0|        3.0|      5.0|
5.0|          7.0|
|                 0.0|          2.0|           2.0|        4.0|      4.0|
1.0|          4.0|
|                 0.0|          2.0|           3.0|        3.0|      5.0|
11.0|          5.0|
|                 0.0|          2.0|           6.0|        3.0|      4.0|
4.0|          9.0|
+--------------------+-------------+--------------+-----------+---------+----
-----+-------------+
only showing top 5 rows
```

[159]:
```
df = df.fillna(0, subset =␣
↪["num_accts_ever_120_pd","num_actv_bc_tl","num_actv_rev_tl","num_bc_sats","num_bc_tl","num_
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 161, Finished, Available,␣
↪Finished)
```

[160]:
```
df.
↪select(["num_rev_accts","num_rev_tl_bal_gt_0","num_sats","num_tl_120dpd_2m","num_tl_30dpd",
↪distinct().show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 162, Finished, Available,␣
↪Finished)
```

```
+-------------+-------------------+--------+---------------+-----------+------
-----------+------------------+
|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|
+-------------+-------------------+--------+---------------+-----------+------
-----------+------------------+
|         31.0|               16.0|    22.0|            0.0|        0.0|
0.0|               2.0|
|         30.0|                7.0|    18.0|            0.0|        0.0|
0.0|               2.0|
|          3.0|                3.0|    10.0|            0.0|        0.0|
0.0|               0.0|
|          8.0|                8.0|    10.0|            0.0|        0.0|
0.0|               1.0|
|         19.0|               11.0|    20.0|            0.0|        0.0|
0.0|               4.0|
+-------------+-------------------+--------+---------------+-----------+------
-----------+------------------+
only showing top 5 rows
```

[161]:
```
df =df.fillna(0, subset =␣
↪["num_rev_accts","num_rev_tl_bal_gt_0","num_sats","num_tl_120dpd_2m","num_tl_30dpd","num_tl
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 163, Finished, Available,␣
↪Finished)
```

[162]:
```
df.
↪select(["pct_tl_nvr_dlq","percent_bc_gt_75","pub_rec_bankruptcies","tax_liens","tot_hi_cred
↪distinct().show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 164, Finished, Available,␣
↪Finished)
```

```
+-------------+---------------+-------------------+---------+--------------+
----------------+--------------+
```

```
|pct_tl_nvr_dlq|percent_bc_gt_75|pub_rec_bankruptcies|tax_liens|tot_hi_cred_lim|
total_bal_ex_mort|total_bc_limit|
+--------------+----------------+--------------------+---------+---------------+
-----------------+--------------+
|         100.0|            62.5|                 0.0|      0.0|       346800.0|
79979.0|       67100.0|
|         100.0|             0.0|                 0.0|      0.0|         1300.0|
0.0|        1000.0|
|         100.0|            66.7|                 0.0|      0.0|        25000.0|
18085.0|       12300.0|
|         100.0|            66.7|                 0.0|      0.0|        48599.0|
40956.0|        6400.0|
|         100.0|            58.3|                 0.0|      0.0|       498462.0|
97740.0|       94400.0|
+--------------+----------------+--------------------+---------+---------------+
-----------------+--------------+
only showing top 5 rows
```

[163]:
```
df=df.fillna(0, subset =␣
  ↪["pct_tl_nvr_dlq","percent_bc_gt_75","pub_rec_bankruptcies","tax_liens","tot_hi_cred_lim","
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 165, Finished, Available,␣
  ↪Finished)
```

[164]:
```
df.
  ↪select(["total_il_high_credit_limit","hardship_flag","disbursement_method","debt_settlement
  ↪distinct().show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 166, Finished, Available,␣
  ↪Finished)

+------------------------+------------+------------------+-----------------
--+
|total_il_high_credit_limit|hardship_flag|disbursement_method|debt_settlement_fl
ag|
+------------------------+------------+------------------+-----------------
--+
|                123139.0|           N|              Cash|
N|
|                 36944.0|           N|              Cash|
N|
|                 21000.0|           N|          DirectPay|
N|
|                 94043.0|           N|              Cash|
Y|
|                 43789.0|           N|              Cash|
Y|
```

```
+-------------------------+------------+------------------+-----------------
--+
only showing top 5 rows
```

[165]:
```
df=df.fillna(0, subset = "total_il_high_credit_limit")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 167, Finished, Available,␣
↪Finished)
```

[166]:
```
df.select("hardship_flag").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 168, Finished, Available,␣
↪Finished)
```

```
+-------------+
|hardship_flag|
+-------------+
|            Y|
|            N|
+-------------+
```

[167]:
```
df.select("disbursement_method").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 169, Finished, Available,␣
↪Finished)
```

```
+-------------------+
|disbursement_method|
+-------------------+
|               Cash|
|          DirectPay|
+-------------------+
```

[168]:
```
df.select("debt_settlement_flag").distinct().show()
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 170, Finished, Available,␣
↪Finished)
```

```
+--------------------+
|debt_settlement_flag|
+--------------------+
|                   Y|
|                   N|
+--------------------+
```

[169]:
```
col_to_drop = ['member_id', 'desc', 'mths_since_last_record',␣
 ↪'annual_inc_joint', 'dti_joint', 'verification_status_joint',␣
 ↪'mths_since_recent_bc_dlq', 'revol_bal_joint', 'sec_app_fico_range_low',␣
 ↪'sec_app_fico_range_high', 'sec_app_earliest_cr_line',␣
 ↪'sec_app_inq_last_6mths', 'sec_app_mort_acc', 'sec_app_open_acc',␣
 ↪'sec_app_revol_util', 'sec_app_open_act_il', 'sec_app_num_rev_accts',␣
 ↪'sec_app_chargeoff_within_12_mths', 'sec_app_collections_12_mths_ex_med',␣
 ↪'sec_app_mths_since_last_major_derog', 'hardship_type', 'hardship_reason',␣
 ↪'hardship_status', 'deferral_term', 'hardship_amount',␣
 ↪'hardship_start_date', 'hardship_end_date', 'payment_plan_start_date',␣
 ↪'hardship_length', 'hardship_dpd', 'hardship_loan_status',␣
 ↪'orig_projected_additional_accrued_interest',␣
 ↪'hardship_payoff_balance_amount', 'hardship_last_payment_amount',␣
 ↪'debt_settlement_flag_date', 'settlement_status', 'settlement_date',␣
 ↪'settlement_amount', 'settlement_percentage',␣
 ↪'settlement_term','Term','emp_length','url','mths_since_last_major_derog']
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 171, Finished, Available,␣
 ↪Finished)
```

[173]:
```
col_to_drop #44
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 175, Finished, Available,␣
 ↪Finished)
```

[173]:
```
['member_id',
 'desc',
 'mths_since_last_record',
 'annual_inc_joint',
 'dti_joint',
 'verification_status_joint',
 'mths_since_recent_bc_dlq',
 'revol_bal_joint',
 'sec_app_fico_range_low',
 'sec_app_fico_range_high',
 'sec_app_earliest_cr_line',
 'sec_app_inq_last_6mths',
 'sec_app_mort_acc',
 'sec_app_open_acc',
 'sec_app_revol_util',
 'sec_app_open_act_il',
 'sec_app_num_rev_accts',
 'sec_app_chargeoff_within_12_mths',
 'sec_app_collections_12_mths_ex_med',
 'sec_app_mths_since_last_major_derog',
 'hardship_type',
 'hardship_reason',
```

```
 'hardship_status',
 'deferral_term',
 'hardship_amount',
 'hardship_start_date',
 'hardship_end_date',
 'payment_plan_start_date',
 'hardship_length',
 'hardship_dpd',
 'hardship_loan_status',
 'orig_projected_additional_accrued_interest',
 'hardship_payoff_balance_amount',
 'hardship_last_payment_amount',
 'debt_settlement_flag_date',
 'settlement_status',
 'settlement_date',
 'settlement_amount',
 'settlement_percentage',
 'settlement_term',
 'Term',
 'emp_length',
 'url',
 'mths_since_last_major_derog']
```

[175]:
```
len(df.columns) #totalcomuns
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 177, Finished, Available,␣
 ↪Finished)
```

[175]: 153

[176]:
```
df=df.drop(*col_to_drop) #dropping the un necessary columns
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 178, Finished, Available,␣
 ↪Finished)
```

[177]:
```
len(df.columns)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 179, Finished, Available,␣
 ↪Finished)
```

[177]: 109

[178]:
```
display(df.head(5))
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 180, Finished, Available,␣
 ↪Finished)
```

```
SynapseWidget(Synapse.DataFrame, 54c912f3-509d-4e1c-919e-04edc387a8ad)
```

[180]:
```
df.select("id").distinct().show(5)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 182, Finished, Available,␣
↪Finished)
```

```
+--------+
|      id|
+--------+
|56622220|
|12597709|
|48324574|
|48110888|
|59103178|
+--------+
only showing top 5 rows
```

[181]:
```
df.groupBy(col("id")).count().show(30)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 183, Finished, Available,␣
↪Finished)
```

```
+--------+-----+
|      id|count|
+--------+-----+
|56622220|    1|
|12597709|    1|
|48324574|    1|
|48110888|    1|
|59103178|    1|
|57971109|    1|
|55152456|    1|
|57005442|    1|
|53464632|    1|
|34592983|    1|
|58150691|    1|
|58370305|    1|
|58693161|    1|
|52818986|    1|
|55991703|    1|
|47551221|    1|
|52748295|    1|
|34412729|    1|
|11408061|    1|
|33391643|    1|
|11655937|    1|
|57883113|    1|
|58060167|    1|
```

```
|57335371|    1|
|56021594|    1|
|58524152|    1|
|10735779|    1|
|10638934|    1|
|35753400|    1|
|36890207|    1|
+--------+-----+
only showing top 30 rows
```

[182]:
```
df.count(),len(df.columns)
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 184, Finished, Available,␣
↪Finished)
```

[182]:
```
(2260668, 109)
```

[196]:
```
df.write.format("delta").saveAsTable("Cleaned_bank_data")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 198, Finished, Available,␣
↪Finished)
```

[195]:
```
df.write.option("header",True).csv("Files/Cleaned_bank_data.csv")
```

```
StatementMeta(, 3cca4f1d-ae58-44c8-ae2a-a16209ab7dbc, 197, Finished, Available,␣
↪Finished)
```

8 Finally Banking Dataset cleaning completed.

9 Rawdatset : Total colmuns - 151, Total Rows - 22,60,701

10 Cleaned dataset : Total colmuns - 109, Total Rows - 22,60,668

11 Newly added columns count - 3

12 columns removed due to nulls more than 75% - 41