In [1]:
```
'''Case Study

Business Objective: Help Lending Club identify members most likely to defaul

Key skills being assessed: Story telling

Overview
The following exercise will provide you with the opportunity to demonstrate 

Please find the details and expectations below.

Given the attached Lending Club data:
1.          Build an useful model, one which will help the business (it’s ok
2.         Create a Deck to be presented to a semi-technical business audier
3.         Business is looking for actionable insights.
•           How to use the model to make to maximize their profit.
•           Please make assumptions when you do not have the necessary infor
4.         Please also share the Jupyter notebook in Python used to create t
'''
```

Out[1]:
```
'Case Study\n \nBusiness Objective: Help Lending Club identify members most
likely to default on their loan\n \nKey skills being assessed: Story tellin
g\n \nOverview\nThe following exercise will provide you with the opportunit
y to demonstrate your understanding and expertise of data science, machine
learning models, and communication skills.\n \nPlease find the details and
expectations below. \n \nGiven the attached Lending Club data:\n1.
Build an useful model, one which will help the business (it’s okay to take
an estimated guess)\n2.         Create a Deck to be presented to a semi-tec
hnical business audience.\n3.         Business is looking for actionable in
sights.\n•           How to use the model to make to maximize their profit.
\n•           Please make assumptions when you do not have the necessary in
formation.\n4.         Please also share the Jupyter notebook in Python use
d to create the model.\n'
```

In [2]:
```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in /opt/anaconda3/lib/python3.11/sit
e-packages (4.6.0)
Requirement already satisfied: numpy>=1.17.0 in /opt/anaconda3/lib/python3.1
1/site-packages (from lightgbm) (1.26.4)
Requirement already satisfied: scipy in /opt/anaconda3/lib/python3.11/site-p
ackages (from lightgbm) (1.11.4)
```

In [49]:
```
import pandas as pd
from datetime import datetime
import numpy as np
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
import shap
from sklearn.model_selection import train_test_split
```

```python
from sklearn.metrics import classification_report, f1_score

from imblearn.pipeline import Pipeline
import numpy as np
from sklearn.model_selection import train_test_split, StratifiedKFold, cross
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression, RidgeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassif
from sklearn.svm import SVC
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
# view all cols
pd.set_option('display.max_columns', None)
```

In [4]:
```python
df1 = pd.read_csv('~/Downloads/Case Study CSS /Lending Club Data - DR_Demo_L
```

In [5]:
```python
df1.describe
```

Out[5]:

```
<bound method NDFrame.describe of          Id  is_bad            emp_title
emp_length home_ownership  \
0         1       0     Time Warner Cable         10       MORTGAGE
1         2       0     Ottawa University          1           RENT
2         3       0         Kennedy Wilson          4           RENT
3         4       0    TOWN OF PLATTEKILL         10       MORTGAGE
4         5       0  Belmont Correctional         10       MORTGAGE
...     ...     ...                   ...        ...            ...
9995   9996       0                 Cabot          5       MORTGAGE
9996   9997       0         Gallant & Wein          1           RENT
9997   9998       0    Weichert, Realtors          8           RENT
9998   9999       0          meadwestvaco          6       MORTGAGE
9999  10000       0        Rehab Alliance          1           RENT

      annual_inc       verification_status pymnt_plan  \
0        50000.0              not verified          n
1        39216.0              not verified          n
2        65000.0              not verified          n
3        57500.0              not verified          n
4        50004.0         VERIFIED - income          n
...          ...                       ...        ...
9995     66250.0         VERIFIED - income          n
9996     26000.0  VERIFIED - income source          n
9997     47831.0              not verified          n
9998     70000.0              not verified          n
9999     70560.0              not verified          n

                                                  Notes         purpose_cat
\
0                                                   NaN             medical
1       Borrower added on 04/14/11 > I will be using...  debt consolidation
2                                                   NaN         credit card
3                                                   NaN  debt consolidation
4     I want to consolidate my debt, pay for a vacat...  debt consolidation
...                                                 ...                 ...
9995                                                NaN             wedding
9996  Borrower added on 08/30/11 > credit cards cons...  debt consolidation
9997  Borrower added on 03/10/10 > My dream is to fi...  debt consolidation
9998                                                NaN      major purchase
9999    Borrower added on 11/09/11 > order to pay ba...         credit card

                                         purpose zip_code addr_state  \
0                                        Medical    766xx         TX
1                     My Debt Consolidation Loan    660xx         KS
2                               AP Personal Loan    916xx         CA
3                        Debt Consolidation Loan    124xx         NY
4                                    consolidate    439xx         OH
...                                          ...      ...        ...
9995                            Scottish Wedding    014xx         MA
9996                                        debt    112xx         NY
9997  Harnessing credit debt for a stable future.    070xx         NJ
9998                                    personal    244xx         VA
9999                             Credit Card Loan    900xx         CA

      debt_to_income  delinq_2yrs earliest_cr_line  inq_last_6mths  \
0              10.87          0.0          12/1/92             0.0
```

```
1              9.15         0.0         11/1/05            2.0
2             11.24         0.0          6/1/70            0.0
3              6.18         1.0          9/1/82            0.0
4             19.03         0.0         10/1/99            4.0
...             ...         ...             ...            ...
9995           9.40         0.0          9/1/01            1.0
9996          20.49         0.0          5/1/00            1.0
9997          24.13         0.0         12/1/89            0.0
9998          16.18         2.0          3/1/99            2.0
9999          16.13         0.0          9/1/00            1.0

      mths_since_last_delinq  mths_since_last_record  open_acc  pub_rec  \
0                        NaN                     NaN      15.0      0.0
1                        NaN                     NaN       4.0      0.0
2                        NaN                     NaN       4.0      0.0
3                       16.0                     NaN       6.0      0.0
4                        NaN                     NaN       8.0      0.0
...                      ...                     ...       ...      ...
9995                     NaN                     NaN       8.0      0.0
9996                    79.0                     NaN       8.0      0.0
9997                     NaN                   111.0       9.0      1.0
9998                    16.0                     NaN       9.0      0.0
9999                    53.0                     NaN      15.0      0.0

      revol_bal  revol_util  total_acc initial_list_status  \
0         12087        12.1       44.0                   f
1         10114        64.0        5.0                   f
2            81         0.6        8.0                   f
3         10030        37.1       23.0                   f
4         10740        40.4       21.0                   f
...         ...         ...        ...                 ...
9995       3656        24.1       10.0                   f
9996       6709        58.9       12.0                   f
9997      11346        60.7       17.0                   f
9998      17157        50.9       27.0                   f
9999       2304        22.6       34.0                   f

      collections_12_mths_ex_med  mths_since_last_major_derog policy_code
0                            0.0                            1         PC4
1                            0.0                            2         PC1
2                            0.0                            3         PC4
3                            0.0                            2         PC2
4                            0.0                            3         PC3
...                          ...                          ...         ...
9995                         0.0                            2         PC3
9996                         0.0                            2         PC3
9997                         0.0                            3         PC3
9998                         0.0                            2         PC3
9999                         0.0                            2         PC5

[10000 rows x 28 columns]>
```

In [57]:
```
df1.shape
```

Out[57]:
```
(10000, 32)
```

In [58]:
```
df1.columns
```

Out[58]:
```
Index(['Id', 'is_bad', 'emp_title', 'emp_length', 'home_ownership',
       'annual_inc', 'verification_status', 'pymnt_plan', 'Notes',
       'purpose_cat', 'purpose', 'zip_code', 'addr_state', 'debt_to_incom
e',
       'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths',
       'mths_since_last_delinq', 'mths_since_last_record', 'open_acc',
       'pub_rec', 'revol_bal', 'revol_util', 'total_acc',
       'initial_list_status', 'collections_12_mths_ex_med',
       'mths_since_last_major_derog', 'policy_code', 'earliest_cr_num_month
s',
       'boolean_list_status', 'bool_pymnt_plan', 'bool_list_status'],
      dtype='object')
```

In [59]:
```
for i in range(df1.shape[1]):
    print(i, pd.isna(df1.iloc[:, i]).sum(), df1.columns[i])
```

```
0 0 Id
1 0 is_bad
2 592 emp_title
3 0 emp_length
4 0 home_ownership
5 1 annual_inc
6 0 verification_status
7 0 pymnt_plan
8 3231 Notes
9 0 purpose_cat
10 4 purpose
11 0 zip_code
12 0 addr_state
13 0 debt_to_income
14 5 delinq_2yrs
15 5 earliest_cr_line
16 5 inq_last_6mths
17 6316 mths_since_last_delinq
18 9160 mths_since_last_record
19 5 open_acc
20 5 pub_rec
21 0 revol_bal
22 26 revol_util
23 5 total_acc
24 0 initial_list_status
25 32 collections_12_mths_ex_med
26 0 mths_since_last_major_derog
27 0 policy_code
28 5 earliest_cr_num_months
29 0 boolean_list_status
30 0 bool_pymnt_plan
31 0 bool_list_status
```

In [7]:
```
for i in range(df1.shape[1]):
    print(i,df1.columns[i],  len(pd.unique(df1.iloc[:, i])))
```

```
0 Id 10000
1 is_bad 2
2 emp_title 8184
3 emp_length 14
4 home_ownership 5
5 annual_inc 1902
6 verification_status 3
7 pymnt_plan 2
8 Notes 6761
9 purpose_cat 27
10 purpose 5678
11 zip_code 720
12 addr_state 50
13 debt_to_income 2585
14 delinq_2yrs 11
15 earliest_cr_line 464
16 inq_last_6mths 21
17 mths_since_last_delinq 92
18 mths_since_last_record 95
19 open_acc 37
20 pub_rec 5
21 revol_bal 8130
22 revol_util 1028
23 total_acc 76
24 initial_list_status 2
25 collections_12_mths_ex_med 2
26 mths_since_last_major_derog 3
27 policy_code 5
```

In [8]:
```
df1.columns
```

Out[8]:
```
Index(['Id', 'is_bad', 'emp_title', 'emp_length', 'home_ownership',
       'annual_inc', 'verification_status', 'pymnt_plan', 'Notes',
       'purpose_cat', 'purpose', 'zip_code', 'addr_state', 'debt_to_incom
e',
       'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths',
       'mths_since_last_delinq', 'mths_since_last_record', 'open_acc',
       'pub_rec', 'revol_bal', 'revol_util', 'total_acc',
       'initial_list_status', 'collections_12_mths_ex_med',
       'mths_since_last_major_derog', 'policy_code'],
      dtype='object')
```

In [9]:
```
# Imbalanced data
df1[df1.loc[: , 'is_bad'] == 0 ].shape , df1[df1.loc[: , 'is_bad'] == 1 ].sh
```

Out[9]:
```
((8705, 28), (1295, 28))
```

In [10]:
```
# Let's take a look at the firts 5 rows in the data.

df1.head()
```

Out[10]:

	Id	is_bad	emp_title	emp_length	home_ownership	annual_inc	verification_stat
0	1	0	Time Warner Cable	10	MORTGAGE	50000.0	not verif
1	2	0	Ottawa University	1	RENT	39216.0	not verif
2	3	0	Kennedy Wilson	4	RENT	65000.0	not verif
3	4	0	TOWN OF PLATTEKILL	10	MORTGAGE	57500.0	not verif
4	5	0	Belmont Correctional	10	MORTGAGE	50004.0	VERIFIED - inco

In [11]:
```
df1.shape[0]
```

Out[11]: 10000

In [12]:
```
df1.loc[1, 'earliest_cr_line']
```

Out[12]: '11/1/05'

In [13]:
```
# This feature is created to see how long it has been since first credit ope
# Holds how many months have been since first credit.
today = datetime.now()
df1['earliest_cr_num_months'] = 0
for i in range(df1.shape[0]):
    start_date = pd.to_datetime(df1.loc[i, 'earliest_cr_line'], format='%m/%
    df1.loc[i, 'earliest_cr_num_months'] = (today.year - start_date.year) *
```

In [60]:
```
df1[['earliest_cr_line', 'earliest_cr_num_months']]
```

Out[60]:

	earliest_cr_line	earliest_cr_num_months
0	12/1/92	395.0
1	11/1/05	240.0
2	6/1/70	665.0
3	9/1/82	518.0
4	10/1/99	313.0
...	...	...
9995	9/1/01	290.0
9996	5/1/00	306.0
9997	12/1/89	431.0
9998	3/1/99	320.0
9999	9/1/00	302.0

10000 rows × 2 columns

In [14]:
```
# Checking each col
pd.unique(df1['home_ownership'])
```

Out[14]:
```
array(['MORTGAGE', 'RENT', 'OWN', 'OTHER', 'NONE'], dtype=object)
```

In [15]:
```
pd.unique(df1['verification_status'])
```

Out[15]:
```
array(['not verified', 'VERIFIED - income', 'VERIFIED - income source'],
      dtype=object)
```

In [16]:
```
pd.unique(df1['pymnt_plan'])
```

Out[16]:
```
array(['n', 'y'], dtype=object)
```

In [17]:
```
df1[df1['initial_list_status']=='m'].shape , df1[df1['initial_list_status']=
```

Out[17]:
```
((17, 29), (9983, 29))
```

In [18]:
```
df1['boolean_list_status'] = 0
df1['bool_pymnt_plan'] = 0
mapping_dict = {'m': True, 'f': False}
# Apply the mapping to the column
df1['bool_list_status'] = df1['initial_list_status'].map(mapping_dict)
mapping_dict = {'y': True, 'n': False}
df1['bool_pymnt_plan'] = df1['pymnt_plan'].map(mapping_dict)
```

In [19]:
```
df2 = pd.get_dummies(df1, columns= ['purpose_cat', 'verification_status', 'h
```

In [20]:
```
df2.columns
```

Out[20]:
```
Index(['Id', 'is_bad', 'emp_title', 'emp_length', 'annual_inc', 'pymnt_pla
n',
       'Notes', 'purpose', 'zip_code', 'addr_state', 'debt_to_income',
       'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths',
       'mths_since_last_delinq', 'mths_since_last_record', 'open_acc',
       'pub_rec', 'revol_bal', 'revol_util', 'total_acc',
       'initial_list_status', 'collections_12_mths_ex_med',
       'mths_since_last_major_derog', 'earliest_cr_num_months',
       'boolean_list_status', 'bool_pymnt_plan', 'bool_list_status',
       'purpose_cat_car', 'purpose_cat_car small business',
       'purpose_cat_credit card', 'purpose_cat_credit card small business',
       'purpose_cat_debt consolidation',
       'purpose_cat_debt consolidation small business',
       'purpose_cat_educational', 'purpose_cat_educational small business',
       'purpose_cat_home improvement',
       'purpose_cat_home improvement small business', 'purpose_cat_house',
       'purpose_cat_house small business', 'purpose_cat_major purchase',
       'purpose_cat_major purchase small business', 'purpose_cat_medical',
       'purpose_cat_medical small business', 'purpose_cat_moving',
       'purpose_cat_moving small business', 'purpose_cat_other',
       'purpose_cat_other small business', 'purpose_cat_renewable energy',
       'purpose_cat_small business',
       'purpose_cat_small business small business', 'purpose_cat_vacation',
       'purpose_cat_vacation small business', 'purpose_cat_wedding',
       'purpose_cat_wedding small business',
       'verification_status_VERIFIED - income',
       'verification_status_VERIFIED - income source',
       'verification_status_not verified', 'home_ownership_MORTGAGE',
       'home_ownership_NONE', 'home_ownership_OTHER', 'home_ownership_OWN',
       'home_ownership_RENT', 'policy_code_PC1', 'policy_code_PC2',
       'policy_code_PC3', 'policy_code_PC4', 'policy_code_PC5'],
      dtype='object')
```

In [21]:
```python
for i in range(df2.shape[1]):
    print(i, pd.isna(df2.iloc[:, i]).sum(), df2.columns[i] )
```

```
0 0 Id
1 0 is_bad
2 592 emp_title
3 0 emp_length
4 1 annual_inc
5 0 pymnt_plan
6 3231 Notes
7 4 purpose
8 0 zip_code
9 0 addr_state
10 0 debt_to_income
11 5 delinq_2yrs
12 5 earliest_cr_line
13 5 inq_last_6mths
14 6316 mths_since_last_delinq
15 9160 mths_since_last_record
16 5 open_acc
17 5 pub_rec
18 0 revol_bal
19 26 revol_util
20 5 total_acc
21 0 initial_list_status
22 32 collections_12_mths_ex_med
23 0 mths_since_last_major_derog
24 5 earliest_cr_num_months
25 0 boolean_list_status
26 0 bool_pymnt_plan
27 0 bool_list_status
28 0 purpose_cat_car
29 0 purpose_cat_car small business
30 0 purpose_cat_credit card
31 0 purpose_cat_credit card small business
32 0 purpose_cat_debt consolidation
33 0 purpose_cat_debt consolidation small business
34 0 purpose_cat_educational
35 0 purpose_cat_educational small business
36 0 purpose_cat_home improvement
37 0 purpose_cat_home improvement small business
38 0 purpose_cat_house
39 0 purpose_cat_house small business
40 0 purpose_cat_major purchase
41 0 purpose_cat_major purchase small business
42 0 purpose_cat_medical
43 0 purpose_cat_medical small business
44 0 purpose_cat_moving
45 0 purpose_cat_moving small business
46 0 purpose_cat_other
47 0 purpose_cat_other small business
48 0 purpose_cat_renewable energy
49 0 purpose_cat_small business
50 0 purpose_cat_small business small business
51 0 purpose_cat_vacation
52 0 purpose_cat_vacation small business
53 0 purpose_cat_wedding
54 0 purpose_cat_wedding small business
55 0 verification_status_VERIFIED - income
```

```
56 0 verification_status_VERIFIED - income source
57 0 verification_status_not verified
58 0 home_ownership_MORTGAGE
59 0 home_ownership_NONE
60 0 home_ownership_OTHER
61 0 home_ownership_OWN
62 0 home_ownership_RENT
63 0 policy_code_PC1
64 0 policy_code_PC2
65 0 policy_code_PC3
66 0 policy_code_PC4
67 0 policy_code_PC5
```

In [22]:
```
df2[pd.isna(df2[ 'delinq_2yrs'])][['delinq_2yrs', 'total_acc', 'open_acc','p
```

Out[22]:

	delinq_2yrs	total_acc	open_acc	pub_rec	inq_last_6mths	revol_util	annual_
4319	NaN	NaN	NaN	NaN	NaN	NaN	18000
4328	NaN	NaN	NaN	NaN	NaN	NaN	5000
4678	NaN	NaN	NaN	NaN	NaN	NaN	600
6232	NaN	NaN	NaN	NaN	NaN	NaN	650
7592	NaN	NaN	NaN	NaN	NaN	NaN	N

In [23]:
```
missing_data_record_indexes = df2[pd.isna(df2[ 'delinq_2yrs'])].index.to_lis
remaining_indices = df2.index.difference(missing_data_record_indexes)
df3 = df2.loc[remaining_indices]
df3 = df3[['is_bad', 'bool_pymnt_plan', 'emp_length','annual_inc', 'boolean_
             'mths_since_last_major_derog','earliest_cr_num_months' , 'open_
             'pub_rec','revol_bal', 'revol_util','inq_last_6mths', 'total_ac
       'purpose_cat_credit card small business', 'purpose_cat_debt consolida
       'purpose_cat_educational', 'purpose_cat_educational small business','
       'purpose_cat_home improvement small business', 'purpose_cat_house','p
       'purpose_cat_major purchase small business', 'purpose_cat_medical','p
       'purpose_cat_moving small business', 'purpose_cat_other','purpose_cat
       'purpose_cat_small business small business', 'purpose_cat_vacation','
       'purpose_cat_wedding small business', 'verification_status_VERIFIED -
       'verification_status_not verified', 'home_ownership_MORTGAGE','home_o
             'home_ownership_RENT', 'policy_code_PC1', 'policy_code_PC2',
       'policy_code_PC3', 'policy_code_PC4', 'policy_code_PC5']]
```

In [24]:
```
for i in range(df3.shape[1]):
    print(i, pd.isna(df3.iloc[:, i]).sum(), df3.columns[i] )
```

```
0 0 is_bad
1 0 bool_pymnt_plan
2 0 emp_length
3 0 annual_inc
4 0 boolean_list_status
5 0 delinq_2yrs
6 0 debt_to_income
7 0 mths_since_last_major_derog
8 0 earliest_cr_num_months
9 0 open_acc
10 0 pub_rec
11 0 revol_bal
12 21 revol_util
13 0 inq_last_6mths
14 0 total_acc
15 0 purpose_cat_car
16 0 purpose_cat_car small business
17 0 purpose_cat_credit card
18 0 purpose_cat_credit card small business
19 0 purpose_cat_debt consolidation
20 0 purpose_cat_debt consolidation small business
21 0 purpose_cat_educational
22 0 purpose_cat_educational small business
23 0 purpose_cat_home improvement
24 0 purpose_cat_home improvement small business
25 0 purpose_cat_house
26 0 purpose_cat_house small business
27 0 purpose_cat_major purchase
28 0 purpose_cat_major purchase small business
29 0 purpose_cat_medical
30 0 purpose_cat_medical small business
31 0 purpose_cat_moving
32 0 purpose_cat_moving small business
33 0 purpose_cat_other
34 0 purpose_cat_other small business
35 0 purpose_cat_renewable energy
36 0 purpose_cat_small business
37 0 purpose_cat_small business small business
38 0 purpose_cat_vacation
39 0 purpose_cat_vacation small business
40 0 purpose_cat_wedding
41 0 purpose_cat_wedding small business
42 0 verification_status_VERIFIED - income
43 0 verification_status_VERIFIED - income source
44 0 verification_status_not verified
45 0 home_ownership_MORTGAGE
46 0 home_ownership_NONE
47 0 home_ownership_OTHER
48 0 home_ownership_OWN
49 0 home_ownership_RENT
50 0 policy_code_PC1
51 0 policy_code_PC2
52 0 policy_code_PC3
53 0 policy_code_PC4
54 0 policy_code_PC5
```

In [25]:
```
df3.head()
```

Out[25]:

	is_bad	bool_pymnt_plan	emp_length	annual_inc	boolean_list_status	delinq_2yr
0	0	False	10	50000.0	0	0.(
1	0	False	1	39216.0	0	0.(
2	0	False	4	65000.0	0	0.(
3	0	False	10	57500.0	0	1.(
4	0	False	10	50004.0	0	0.(

In [26]:
```
df3.shape
```

Out[26]: (9995, 55)

In [27]:
```
df3 = df3.fillna(0)
```

In [28]:
```
for i in range(df3.shape[1]):
    print(i, pd.isna(df3.iloc[:, i]).sum(), df3.columns[i] )
```

```
0 0 is_bad
1 0 bool_pymnt_plan
2 0 emp_length
3 0 annual_inc
4 0 boolean_list_status
5 0 delinq_2yrs
6 0 debt_to_income
7 0 mths_since_last_major_derog
8 0 earliest_cr_num_months
9 0 open_acc
10 0 pub_rec
11 0 revol_bal
12 0 revol_util
13 0 inq_last_6mths
14 0 total_acc
15 0 purpose_cat_car
16 0 purpose_cat_car small business
17 0 purpose_cat_credit card
18 0 purpose_cat_credit card small business
19 0 purpose_cat_debt consolidation
20 0 purpose_cat_debt consolidation small business
21 0 purpose_cat_educational
22 0 purpose_cat_educational small business
23 0 purpose_cat_home improvement
24 0 purpose_cat_home improvement small business
25 0 purpose_cat_house
26 0 purpose_cat_house small business
27 0 purpose_cat_major purchase
28 0 purpose_cat_major purchase small business
29 0 purpose_cat_medical
30 0 purpose_cat_medical small business
31 0 purpose_cat_moving
32 0 purpose_cat_moving small business
33 0 purpose_cat_other
34 0 purpose_cat_other small business
35 0 purpose_cat_renewable energy
36 0 purpose_cat_small business
37 0 purpose_cat_small business small business
38 0 purpose_cat_vacation
39 0 purpose_cat_vacation small business
40 0 purpose_cat_wedding
41 0 purpose_cat_wedding small business
42 0 verification_status_VERIFIED - income
43 0 verification_status_VERIFIED - income source
44 0 verification_status_not verified
45 0 home_ownership_MORTGAGE
46 0 home_ownership_NONE
47 0 home_ownership_OTHER
48 0 home_ownership_OWN
49 0 home_ownership_RENT
50 0 policy_code_PC1
51 0 policy_code_PC2
52 0 policy_code_PC3
53 0 policy_code_PC4
54 0 policy_code_PC5
```

In [29]:

```
for i in range(df3.shape[1]):
    print(i, pd.isna(df3.iloc[:, i]).sum(), df3.columns[i] )
```

```
0 0 is_bad
1 0 bool_pymnt_plan
2 0 emp_length
3 0 annual_inc
4 0 boolean_list_status
5 0 delinq_2yrs
6 0 debt_to_income
7 0 mths_since_last_major_derog
8 0 earliest_cr_num_months
9 0 open_acc
10 0 pub_rec
11 0 revol_bal
12 0 revol_util
13 0 inq_last_6mths
14 0 total_acc
15 0 purpose_cat_car
16 0 purpose_cat_car small business
17 0 purpose_cat_credit card
18 0 purpose_cat_credit card small business
19 0 purpose_cat_debt consolidation
20 0 purpose_cat_debt consolidation small business
21 0 purpose_cat_educational
22 0 purpose_cat_educational small business
23 0 purpose_cat_home improvement
24 0 purpose_cat_home improvement small business
25 0 purpose_cat_house
26 0 purpose_cat_house small business
27 0 purpose_cat_major purchase
28 0 purpose_cat_major purchase small business
29 0 purpose_cat_medical
30 0 purpose_cat_medical small business
31 0 purpose_cat_moving
32 0 purpose_cat_moving small business
33 0 purpose_cat_other
34 0 purpose_cat_other small business
35 0 purpose_cat_renewable energy
36 0 purpose_cat_small business
37 0 purpose_cat_small business small business
38 0 purpose_cat_vacation
39 0 purpose_cat_vacation small business
40 0 purpose_cat_wedding
41 0 purpose_cat_wedding small business
42 0 verification_status_VERIFIED - income
43 0 verification_status_VERIFIED - income source
44 0 verification_status_not verified
45 0 home_ownership_MORTGAGE
46 0 home_ownership_NONE
47 0 home_ownership_OTHER
48 0 home_ownership_OWN
49 0 home_ownership_RENT
50 0 policy_code_PC1
51 0 policy_code_PC2
52 0 policy_code_PC3
53 0 policy_code_PC4
54 0 policy_code_PC5
```

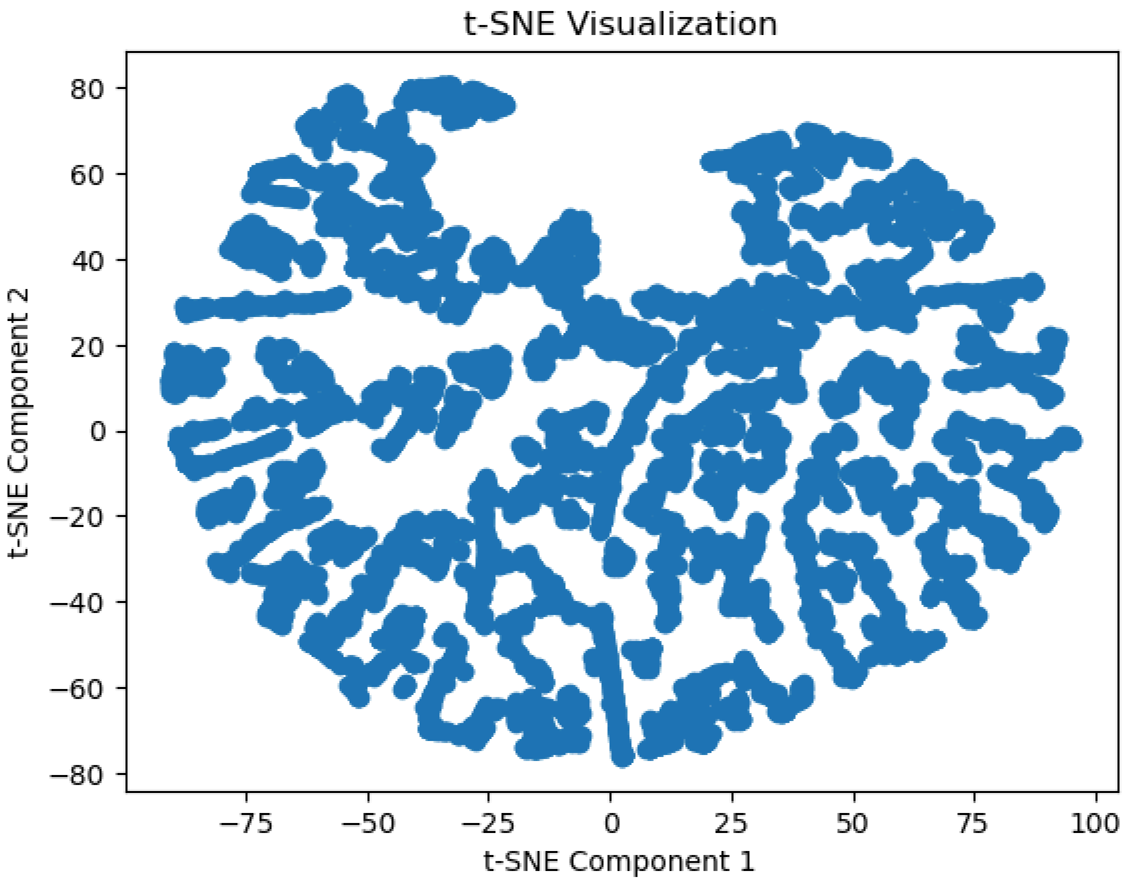
In [30]:
```
X = df3[[ 'bool_pymnt_plan', 'emp_length','annual_inc', 'total_acc', 'boolea
             'mths_since_last_major_derog','earliest_cr_num_months' , 'open_
             'pub_rec','revol_bal', 'revol_util','inq_last_6mths','purpose_c
       'purpose_cat_credit card small business', 'purpose_cat_debt consolida
       'purpose_cat_educational', 'purpose_cat_educational small business','
       'purpose_cat_home improvement small business', 'purpose_cat_house','p
       'purpose_cat_major purchase small business', 'purpose_cat_medical','p
       'purpose_cat_moving small business', 'purpose_cat_other','purpose_cat
       'purpose_cat_small business small business', 'purpose_cat_vacation','
       'purpose_cat_wedding small business', 'verification_status_VERIFIED -
       'verification_status_not verified', 'home_ownership_MORTGAGE','home_c
             'home_ownership_RENT', 'policy_code_PC1', 'policy_code_PC2',
       'policy_code_PC3', 'policy_code_PC4', 'policy_code_PC5']]
y = df3['is_bad']
```

In [64]:
```
X.shape
```

Out[64]: (9995, 54)

In [62]:
```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
tsne = TSNE(n_components=2, random_state=42, perplexity=30)
X_embedded = tsne.fit_transform(X)
plt.scatter(X_embedded[:, 0], X_embedded[:, 1])
plt.title("t-SNE Visualization")
plt.xlabel("t-SNE Component 1")
plt.ylabel("t-SNE Component 2")
```

Out[62]: Text(0, 0.5, 't-SNE Component 2')

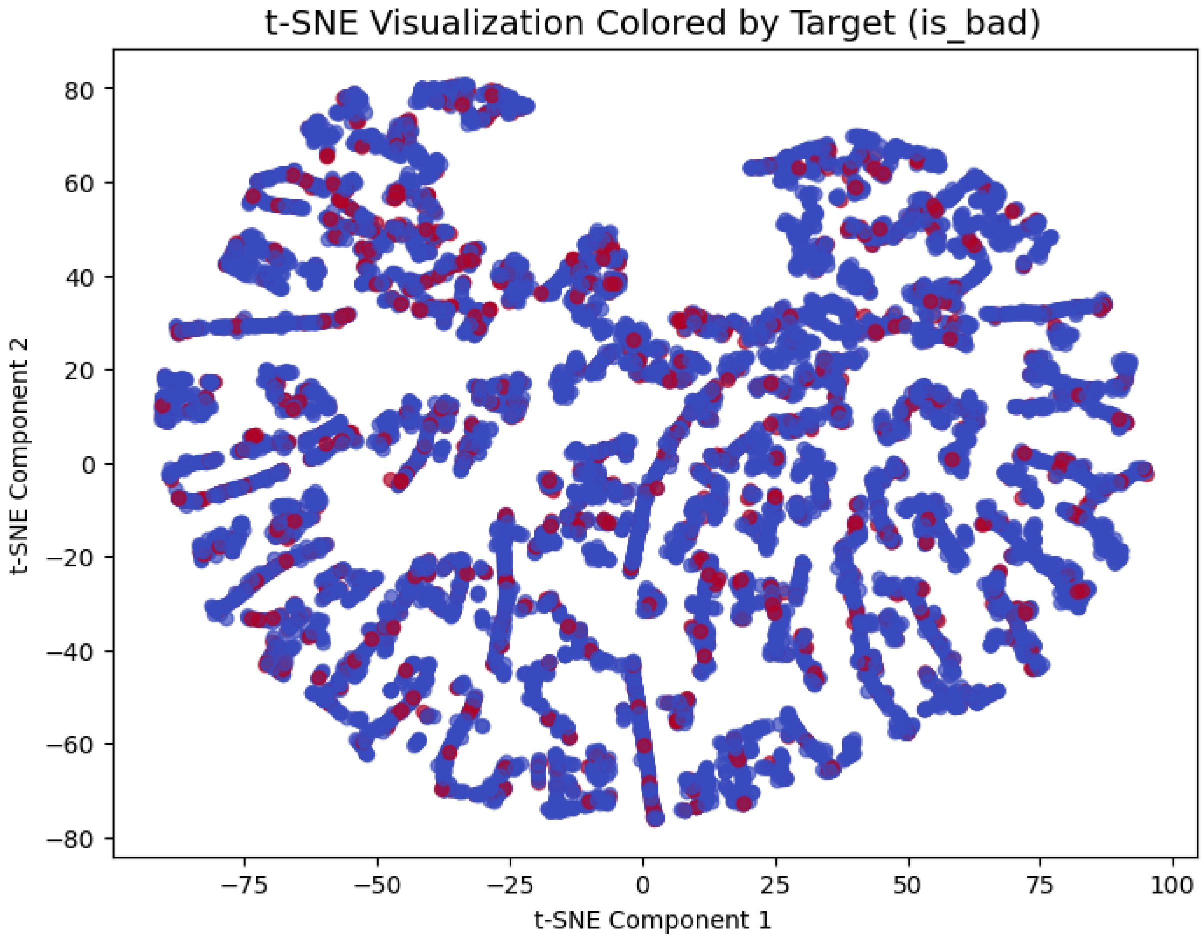


In [63]:
```
tsne = TSNE(n_components=2, random_state=42, perplexity=30)
X_embedded = tsne.fit_transform(X)

# Create the scatter plot with target-based colors
plt.figure(figsize=(8, 6))
scatter = plt.scatter(
    X_embedded[:, 0],
    X_embedded[:, 1],
    c=y,                        # color by target
    cmap='coolwarm',            # red-blue color map
    alpha=0.7,                  # transparency for better visibility
    s=30                        # marker size
)

# Add labels and title
plt.title("t-SNE Visualization Colored by Target (is_bad)", fontsize=14)
plt.xlabel("t-SNE Component 1")
plt.ylabel("t-SNE Component 2")
```

Out[63]:
```
Text(0, 0.5, 't-SNE Component 2')
```



In [ ]:

In [32]:
```
y[y == 0 ].shape , y[y==1].shape
```

Out[32]: ((8700,), (1295,))

In [33]:
```
X = X.apply(lambda x: x.astype(float) if x.dtype == bool else x)
```

In [34]:
```
X[X['emp_length']=='na']
```

Out[34]:

	bool_pymnt_plan	emp_length	annual_inc	total_acc	boolean_list_status	delir
119	0.0	na	55200.0	53.0	0	
255	0.0	na	63000.0	16.0	0	
404	0.0	na	12360.0	11.0	0	
414	0.0	na	30000.0	19.0	0	
472	0.0	na	30000.0	6.0	0	
...	...	...	...	...	...	
9698	0.0	na	38376.0	16.0	0	
9724	0.0	na	25980.0	21.0	0	
9732	0.0	na	35500.0	52.0	0	
9960	0.0	na	65000.0	29.0	0	
9962	0.0	na	50400.0	17.0	0	

250 rows × 54 columns

In [35]:

```
def clean_emp_length(val):
    if pd.isna(val):
        return np.nan
    val = str(val).lower().strip()
    digits = ''.join([c for c in val if c.isdigit()])
    return float(digits) if digits else np.nan
    # Step 1: Convert to numeric float
X['emp_length'] = X['emp_length'].apply(clean_emp_length)

# Step 2: Fill missing/invalid values with 1.0
X['emp_length'] = X['emp_length'].fillna(1.0)

# Step 3: Ensure dtype is float
X['emp_length'] = X['emp_length'].astype(float)
```

In [36]:

```
for i in range(X.shape[1]):
    print(X.iloc[:, i]  )
```

```
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: bool_pymnt_plan, Length: 9995, dtype: float64
0       10.0
1        1.0
2        4.0
3       10.0
4       10.0
        ...
9995     5.0
9996     1.0
9997     8.0
9998     6.0
9999     1.0
Name: emp_length, Length: 9995, dtype: float64
0       50000.0
1       39216.0
2       65000.0
3       57500.0
4       50004.0
         ...
9995    66250.0
9996    26000.0
9997    47831.0
9998    70000.0
9999    70560.0
Name: annual_inc, Length: 9995, dtype: float64
0       44.0
1        5.0
2        8.0
3       23.0
4       21.0
        ...
9995    10.0
9996    12.0
9997    17.0
9998    27.0
9999    34.0
Name: total_acc, Length: 9995, dtype: float64
0       0
1       0
2       0
3       0
4       0
       ..
9995    0
9996    0
```

```
9997    0
9998    0
9999    0
Name: boolean_list_status, Length: 9995, dtype: int64
0       0.0
1       0.0
2       0.0
3       1.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    2.0
9999    0.0
Name: delinq_2yrs, Length: 9995, dtype: float64
0       10.87
1        9.15
2       11.24
3        6.18
4       19.03
        ...
9995     9.40
9996    20.49
9997    24.13
9998    16.18
9999    16.13
Name: debt_to_income, Length: 9995, dtype: float64
0       1
1       2
2       3
3       2
4       3
       ..
9995    2
9996    2
9997    3
9998    2
9999    2
Name: mths_since_last_major_derog, Length: 9995, dtype: int64
0       395.0
1       240.0
2       665.0
3       518.0
4       313.0
        ...
9995    290.0
9996    306.0
9997    431.0
9998    320.0
9999    302.0
Name: earliest_cr_num_months, Length: 9995, dtype: float64
0       15.0
1        4.0
2        4.0
3        6.0
```

```
4        8.0
        ...
9995     8.0
9996     8.0
9997     9.0
9998     9.0
9999    15.0
Name: open_acc, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    1.0
9998    0.0
9999    0.0
Name: pub_rec, Length: 9995, dtype: float64
0       12087
1       10114
2          81
3       10030
4       10740
        ...
9995     3656
9996     6709
9997    11346
9998    17157
9999     2304
Name: revol_bal, Length: 9995, dtype: int64
0       12.1
1       64.0
2        0.6
3       37.1
4       40.4
        ...
9995    24.1
9996    58.9
9997    60.7
9998    50.9
9999    22.6
Name: revol_util, Length: 9995, dtype: float64
0       0.0
1       2.0
2       0.0
3       0.0
4       4.0
       ...
9995    1.0
9996    1.0
9997    0.0
9998    2.0
9999    1.0
Name: inq_last_6mths, Length: 9995, dtype: float64
```

```
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_car, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_car small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       1.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    1.0
Name: purpose_cat_credit card, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_credit card small business, Length: 9995, dtype: float64
0       0.0
1       1.0
2       0.0
3       1.0
4       1.0
       ...
9995    0.0
9996    1.0
```

```
9997    1.0
9998    0.0
9999    0.0
Name: purpose_cat_debt consolidation, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_debt consolidation small business, Length: 9995, dtype: fl
oat64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_educational, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_educational small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_home improvement, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
```

```
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_home improvement small business, Length: 9995, dtype: floa
t64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_house, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_house small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    1.0
9999    0.0
Name: purpose_cat_major purchase, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
```

```
9999    0.0
Name: purpose_cat_major purchase small business, Length: 9995, dtype: float6
4
0       1.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_medical, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_medical small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_moving, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_moving small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
```

```
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_other, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_other small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_renewable energy, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_small business small business, Length: 9995, dtype: float6
4
```

```
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_vacation, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_vacation small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    1.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_wedding, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: purpose_cat_wedding small business, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       1.0
       ...
9995    1.0
9996    0.0
```

```
9997    0.0
9998    0.0
9999    0.0
Name: verification_status_VERIFIED - income, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    1.0
9997    0.0
9998    0.0
9999    0.0
Name: verification_status_VERIFIED - income source, Length: 9995, dtype: flo
at64
0       1.0
1       1.0
2       1.0
3       1.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    1.0
9998    1.0
9999    1.0
Name: verification_status_not verified, Length: 9995, dtype: float64
0       1.0
1       0.0
2       0.0
3       1.0
4       1.0
       ...
9995    1.0
9996    0.0
9997    0.0
9998    1.0
9999    0.0
Name: home_ownership_MORTGAGE, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: home_ownership_NONE, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
```

```
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: home_ownership_OTHER, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: home_ownership_OWN, Length: 9995, dtype: float64
0       0.0
1       1.0
2       1.0
3       0.0
4       0.0
       ...
9995    0.0
9996    1.0
9997    1.0
9998    0.0
9999    1.0
Name: home_ownership_RENT, Length: 9995, dtype: float64
0       0.0
1       1.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: policy_code_PC1, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       1.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
```

```
Name: policy_code_PC2, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       1.0
       ...
9995    1.0
9996    1.0
9997    1.0
9998    1.0
9999    0.0
Name: policy_code_PC3, Length: 9995, dtype: float64
0       1.0
1       0.0
2       1.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    0.0
Name: policy_code_PC4, Length: 9995, dtype: float64
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
9995    0.0
9996    0.0
9997    0.0
9998    0.0
9999    1.0
Name: policy_code_PC5, Length: 9995, dtype: float64
```

In [37]:
```
X['revol_bal'] = X['revol_bal'].astype(float)
```

In [38]:
```
X.shape , y.shape
```

Out[38]:
```
((9995, 54), (9995,))
```

In [39]:
```
# Split data into training testing and validation
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, ra
```

In [40]:
```
X_train.shape , X_test.shape
```

Out[40]:
```
((6996, 54), (2999, 54))
```

In [41]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

```python
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [42]:
```python
X_train_scaled.shape , X_test_scaled.shape
```

Out[42]: ((6996, 54), (2999, 54))

In [43]:
```python
smote = SMOTE(random_state=42)
X_train_bal, y_train_bal = smote.fit_resample(X_train_scaled, y_train)
```

In [44]:
```python
X_train_bal.shape
```

Out[44]: (12180, 54)

In [45]:
```python
from sklearn.ensemble import RandomForestClassifier
clf_rf = RandomForestClassifier(random_state=42)
clf_rf.fit(X_train_bal, y_train_bal)
```

Out[45]:
```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

In [51]:
```python
models = {
    'LogisticRegression': LogisticRegression(max_iter=5000),
    'RidgeClassifier': RidgeClassifier(),
    'RandomForest': RandomForestClassifier(n_estimators=200, max_depth=None,
    'GradientBoosting': GradientBoostingClassifier(n_estimators=200, learnir
    'AdaBoost': AdaBoostClassifier(n_estimators=100, learning_rate=1.0, ranc
    'SVM': SVC(C=1, kernel='rbf', probability=True),
    'XGBoost': XGBClassifier(n_estimators=200, max_depth=5, learning_rate=0.
    'LightGBM': LGBMClassifier(n_estimators=200, num_leaves=31, learning_rat
}
```

In [52]:
```python
test_results = {}
for name, model in models.items():
    print(f"\n Training {name}...")
    pipe = Pipeline([
        ('scaler', StandardScaler()),
        ('model', model)
    ])

    # Fit on full training set
    pipe.fit(X_train, y_train)

    # Predict on validation set
    y_test_pred = pipe.predict(X_test)
    f1 = f1_score(y_test, y_test_pred)
    test_results[name] = (pipe, f1)
    print(f"{name} → Validation F1 = {f1:.3f}")
```

```
 Training LogisticRegression...
LogisticRegression → Validation F1 = 0.267

 Training RidgeClassifier...
RidgeClassifier → Validation F1 = 0.260

 Training RandomForest...
RandomForest → Validation F1 = 0.236

 Training GradientBoosting...
GradientBoosting → Validation F1 = 0.297

 Training AdaBoost...
AdaBoost → Validation F1 = 0.292

 Training SVM...
SVM → Validation F1 = 0.252

 Training XGBoost...
XGBoost → Validation F1 = 0.285

 Training LightGBM...
[LightGBM] [Info] Number of positive: 906, number of negative: 6090
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of te
sting was 0.000607 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1501
[LightGBM] [Info] Number of data points in the train set: 6996, number of us
ed features: 38
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.129503 -> initscore=-1.905
364
[LightGBM] [Info] Start training from score -1.905364
LightGBM → Validation F1 = 0.230
```

In [56]:

```
best_model_name = max(test_results, key=lambda k: test_results[k][1])
best_pipe = test_results[best_model_name][0]

print(f"\n Best Model on Test: {best_model_name}")

y_test_pred = best_pipe.predict(X_test)
print("\nFinal Test Results:")
print(classification_report(y_test, y_test_pred))
```

```
 Best Model on Test: GradientBoosting

Final Test Results:
              precision    recall  f1-score   support

           0       0.89      1.00      0.94      2610
           1       0.91      0.18      0.30       389

    accuracy                           0.89      2999
   macro avg       0.90      0.59      0.62      2999
weighted avg       0.89      0.89      0.86      2999
```

In [55]:
```python
from sklearn.metrics import roc_auc_score, roc_curve

#  Compute AUC for all models on validation

for name, (pipe, f1) in test_results.items():
    # Use probabilities for AUC
    if hasattr(pipe, "predict_proba"):
        y_test_prob = pipe.predict_proba(X_test)[:, 1]
    else:
        # For SVM without probability=True
        y_val_prob = pipe.decision_function(X_test)
    auc = roc_auc_score(y_test, y_test_prob)
    print(f"{name} → Validation AUC = {auc:.3f}")
```

```
LogisticRegression → Validation AUC = 0.698
RidgeClassifier → Validation AUC = 0.698
RandomForest → Validation AUC = 0.688
GradientBoosting → Validation AUC = 0.700
AdaBoost → Validation AUC = 0.701
SVM → Validation AUC = 0.627
XGBoost → Validation AUC = 0.684
LightGBM → Validation AUC = 0.677
```

In [74]:
```python
importances = pd.Series(clf_rf.feature_importances_, index=X_train.columns)
importances.sort_values(ascending=False).head(20)
```

Out[74]:
```
inq_last_6mths                                  0.127160
mths_since_last_major_derog                     0.091324
emp_length                                      0.068218
revol_util                                      0.066540
annual_inc                                      0.059367
total_acc                                       0.058753
revol_bal                                       0.058376
open_acc                                        0.056218
earliest_cr_num_months                          0.055099
debt_to_income                                  0.054619
verification_status_not verified                0.027056
purpose_cat_debt consolidation small business   0.024113
verification_status_VERIFIED - income           0.021528
home_ownership_RENT                             0.021268
home_ownership_MORTGAGE                         0.020765
delinq_2yrs                                     0.020246
verification_status_VERIFIED - income source    0.015577
policy_code_PC2                                 0.014577
purpose_cat_debt consolidation                  0.014099
policy_code_PC3                                 0.013886
dtype: float64
```

In [3]:
```python
import nbjupyter nbconvert --to pdf your_notebook.ipynb
```

```
  Cell In[3], line 1
    jupyter nbconvert --to pdf your_notebook.ipynb
            ^
SyntaxError: invalid syntax
```

In [ ]: