Import Required Dependencies

Tensorflow Decision Forests

In [ ]:
```
from IPython import display
!pip install tensorflow_decision_forests
display.clear_output()
```

Install PySpark

In [ ]:
```
from IPython import display
!sudo apt update
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://dlcdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
!tar xf spark-3.2.1-bin-hadoop3.2.tgz
!pip install -q findspark
!pip install pyspark
!pip install py4j

import os
import sys
import findspark
findspark.init()
findspark.find()

os.environ['PYSPARK_PYTHON'] = sys.executable
os.environ['PYSPARK_DRIVER_PYTHON'] = sys.executable
display.clear_output()
```

Libraries

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

In [ ]:
```
import pyspark
from pyspark.ml import Pipeline
from pyspark.ml.linalg import DenseVector
from pyspark.ml.feature import VectorAssembler, Imputer, StringIndexer, OneHotEncoder
from pyspark.ml.classification import RandomForestClassifier, MultilayerPerceptronClassifier
from pyspark.ml.stat import Correlation

from pyspark.sql.types import IntegerType, DoubleType
import pyspark.sql.functions as F
from pyspark.sql.functions import col, count, lit, udf
from pyspark.sql import SparkSession

from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
```

In [ ]:
```
import tensorflow as tf
import tensorflow_decision_forests as tfdf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from keras.callbacks import EarlyStopping
from sklearn.model_selection import GridSearchCV

import numpy as np
import pandas as pd

#from keras.wrappers.scikit_learn import KerasClassifier
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score, roc_auc_score, confusion_matrix, classificat
```

Stage1-Preprocessing

In [ ]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, count, when, isnan, isnull

spark = SparkSession.builder.getOrCreate()

# Load data into a DataFrame
```

```
df = spark.read.csv("/content/drive/MyDrive/UOW/316/316-project/data/data.csv", header=True, inferSchema=True)
df.show()  # Display the first few records
```

```
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+-------------
------+---------+-------------+---------+------------------+---------+-------------------+-----------
------+------------------+--------+----------+-----+----------+---------------+--------------+--------------------
-+--------------------+--------+-------+---------+---------+---------+------------------+---------+--------------+--
--------+--------------+--------------+------------+------------------+---------+--------------------+---------
---+--------------+------------+-----------------+------------------------+-------------------------+-----------+
---------------+---------------+---------+------------------------+-------------+------------+-----------+---------
--+----------+-----------+-----------+-----------------+-----------+-------+-----------+-----------+----------+------
--+----------------+------+-----------+------------+-----------+
|     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|pymnt_plan|                desc|           p
urpose|               title|zip_code|addr_state|  dti|delinq_2yrs|earliest_cr_line|inq_last_6mths|mths_since_last_delin
q|mths_since_last_record|open_acc|pub_rec|revol_bal|revol_util|total_acc|initial_list_status|out_prncp|out_prncp_inv|to
tal_pymnt|total_pymnt_inv|total_rec_prncp|total_rec_int|total_rec_late_fee|recoveries|collection_recovery_fee|last_pymn
t_d|last_pymnt_amnt|next_pymnt_d|last_credit_pull_d|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_il_6m|open_il_12m|open_il_24m|mths_since_rcnt_il|total_bal_il|il_util|open_rv_12m|open_rv_24m|max_bal_bc|all_ut
il|total_rev_hi_lim|inq_fi|total_cu_tl|inq_last_12m|default_ind|
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+-------------
------+---------+-------------+---------+------------------+---------+-------------------+-----------
------+------------------+--------+----------+-----+----------+---------------+--------------+--------------------
-+--------------------+--------+-------+---------+---------+---------+------------------+---------+--------------+--
--------+--------------+--------------+------------+------------------+---------+--------------------+---------
---+--------------+------------+-----------------+------------------------+-------------------------+-----------+
---------------+---------------+---------+------------------------+-------------+------------+-----------+---------
--+----------+-----------+-----------+-----------------+-----------+-------+-----------+-----------+----------+------
--+----------------+------+-----------+------------+-----------+
|1077501|  1296599|     5000|       5000|         4975.0| 36 months|   10.65|     162.87|    B|       B2|
NULL| 10+ years|          RENT|     24000|           Verified|01-12-2011|         n|  Borrower added ...|       credit_
card|            Computer|   860xx|        AZ|27.65|          0|      01-01-1985|             1|                  NULL|
NULL|       3|      0|    13648|      83.7|        9|                  f|        0|           0|5861.071414|        58
31.78|          5000|       861.07|                0|         0|                     0|  01-01-2015|        171.62|
NULL|        01-01-2016|                           0|                         NULL|          1|      INDIVIDUAL|
NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          0|
|1077430|  1314167|     2500|       2500|         2500.0| 60 months|   15.27|      59.83|    C|       C4|
Ryder|  < 1 year|          RENT|     30000|    Source Verified|01-12-2011|         n|  Borrower added ...|
car|                bike|   309xx|        GA|    1|          0|      01-04-1999|             5|                  NULL|
NULL|       3|      0|     1687|       9.4|        4|                  f|        0|           0|    1008.71|        10
08.71|         456.46|       435.17|                0|    117.08|                     1.11|  01-04-2013|        119.66|
NULL|        01-09-2013|                           0|                         NULL|          1|      INDIVIDUAL|
NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          1|
```

```
|1077175|  1313524|     2400|       2400|         2400.0| 36 months|   15.96|      84.33|    C|       C5|
NULL| 10+ years|          RENT|     12252|       Not Verified|01-12-2011|         n|                NULL|    small_busi
ness|real estate business|   606xx|        IL| 8.72|          0|      01-11-2001|             2|                    NULL|
NULL|       2|      0|     2956|      98.5|       10|                  f|        0|            0|3003.653644|        30
03.65|           2400|       603.65|                  0|         0|                     0|  01-06-2014|         649.91|
NULL|        01-01-2016|                           0|                        NULL|          1|      INDIVIDUAL|
NULL|     NULL|                        NULL|             0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          0|
|1076863|  1277178|    10000|      10000|        10000.0| 36 months|   13.49|     339.31|    C|       C1| AIR RESOURCES
BOARD| 10+ years|          RENT|     49200|    Source Verified|01-12-2011|         n|  Borrower added ...|
other|            personel|   917xx|        CA|   20|          0|      01-02-1996|             1|                    35
|                NULL|      10|      0|     5598|       21|       37|                  f|        0|             0|122
26.30221|        12226.3|          10000|      2209.33|             16.97|         0|                       0|  01-01-20
15|         357.48|        NULL|        01-01-2015|                           0|                       NULL|          1|
INDIVIDUAL|            NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1075358|  1311748|     3000|       3000|         3000.0| 60 months|   12.69|      67.79|    B|       B5|University Med
ica...|    1 year|          RENT|     80000|    Source Verified|01-12-2011|         n|  Borrower added ...|
other|             Personal|   972xx|        OR|17.94|          0|      01-01-1996|             0|                    38
|                NULL|      15|      0|    27783|      53.9|       38|                  f|    766.9|        766.9|
3242.17|        3242.17|         2233.1|      1009.07|                  0|         0|                       0|  01-01-201
6|          67.79|  01-02-2016|        01-01-2016|                           0|                       NULL|          1|
INDIVIDUAL|            NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1075269|  1311441|     5000|       5000|         5000.0| 36 months|     7.9|     156.46|    A|       A4|Veolia Transpo
rtaton|   3 years|          RENT|     36000|    Source Verified|01-12-2011|         n|                NULL|           w
edding|My wedding loan I...|   852xx|        AZ| 11.2|          0|      01-11-2004|             3|                   NUL
L|                NULL|       9|      0|     7963|      28.3|       12|                  f|        0|            0|56
31.377753|        5631.38|           5000|       631.38|                  0|         0|                       0|  01-01-2
015|         161.03|        NULL|        01-09-2015|                           0|                       NULL|          1|
INDIVIDUAL|            NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1069639|  1304742|     7000|       7000|         7000.0| 60 months|   15.96|     170.08|    C|       C5|Southern Star
Pho...|   8 years|          RENT|     47004|       Not Verified|01-12-2011|         n|  Borrower added ...|debt_consoli
dation|                Loan|   280xx|        NC|23.51|          0|      01-07-2005|             1|                    NUL
L|               NULL|       7|      0|    17726|      85.6|       11|                  f|  1889.15|      1889.15|
8136.84|        8136.84|        5110.85|      3025.99|                  0|         0|                       0|  01-01-201
6|         170.08|  01-02-2016|        01-01-2016|                           0|                       NULL|          1|
INDIVIDUAL|            NULL|     NULL|                        NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1072053|  1288686|     3000|       3000|         3000.0| 36 months|   18.64|     109.43|    E|       E1|     MKC Accou
nting |   9 years|          RENT|     48000|    Source Verified|01-12-2011|         n|  Borrower added ...|
```

```
car|     Car Downpayment|   900xx|        CA| 5.35|          0|      01-01-2007|             2|                  NULL|
NULL|       4|      0|     8221|      87.5|        4|                  f|        0|            0|3938.144334|        39
38.14|          3000|       938.14|                  0|         0|                     0|  01-01-2015|         111.34|
NULL|        01-12-2014|                           0|                        NULL|          1|      INDIVIDUAL|
NULL|     NULL|                     NULL|             0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          0|
|1071795|  1306957|     5600|       5600|         5600.0| 60 months|   21.28|     152.39|    F|       F2|
NULL|   4 years|          OWN|     40000|    Source Verified|01-12-2011|         n|  Borrower added ...|    small_busi
ness|Expand Business &...|   958xx|        CA| 5.55|          0|      01-04-2004|             2|                  NULL|
NULL|      11|      0|     5210|      32.6|       13|                  f|        0|            0|     646.02|         6
46.02|         162.02|       294.94|                  0|    189.06|                  2.09|  01-04-2012|         152.39|
NULL|        01-08-2012|                           0|                        NULL|          1|      INDIVIDUAL|
NULL|     NULL|                     NULL|             0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          1|
|1071570|  1306721|     5375|       5375|         5350.0| 60 months|   12.69|     121.45|    B|       B5|           Sta
rbucks|  < 1 year|          RENT|     15000|           Verified|01-12-2011|         n|  Borrower added ...|
other|Building my credi...|   774xx|        TX|18.08|          0|      01-09-2004|             0|                  NULL
|                  NULL|       2|      0|     9279|      36.5|        3|                  f|        0|            0|
1476.19|        1469.34|         673.48|       533.42|                  0|    269.29|                  2.52|  01-11-201
2|         121.45|        NULL|        01-03-2013|                           0|                        NULL|          1|
INDIVIDUAL|            NULL|     NULL|                     NULL|             0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          1|
|1070078|  1305201|     6500|       6500|         6500.0| 60 months|   14.65|     153.45|    C|       C3|Southwest Rura
l m...|   5 years|          OWN|     72000|       Not Verified|01-12-2011|         n|  Borrower added ...|debt_consoli
dation|High intrest Cons...|   853xx|        AZ|16.12|          0|      01-01-1998|             2|                  NUL
L|                  NULL|      14|      0|     4032|      20.6|       23|                  f|        0|            0|
7677.52|        7677.52|           6500|     1177.52|                  0|         0|                     0|  01-06-201
3|        1655.54|        NULL|        01-12-2015|                           0|                        NULL|          1|
INDIVIDUAL|            NULL|     NULL|                     NULL|             0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1069908|  1305008|    12000|      12000|        12000.0| 36 months|   12.69|     402.54|    B|       B5|
UCLA| 10+ years|          OWN|     75000|    Source Verified|01-12-2011|         n|                NULL|debt_consolida
tion|       Consolidation|   913xx|        CA|10.78|          0|      01-10-1989|             0|                  NULL|
NULL|      12|      0|    23336|      67.1|       34|                  f|        0|            0|   13943.08|       139
43.08|          12000|      1943.08|                  0|         0|                     0|  01-09-2013|         6315.3|
NULL|        01-08-2013|                           0|                        NULL|          1|      INDIVIDUAL|
NULL|     NULL|                     NULL|             0|        NULL|       NULL|       NULL|      NULL|       NULL|
NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|            NULL|  NULL|
NULL|        NULL|          0|
|1064687|  1298717|     9000|       9000|         9000.0| 36 months|   13.49|     305.38|    C|       C1|Va. Dept of Co
nse...|  < 1 year|          RENT|     30000|    Source Verified|01-12-2011|         n|  Borrower added ...|debt_consoli
dation|             freedom|   245xx|        VA|10.08|          0|      01-04-2004|             1|                  NUL
L|                  NULL|       4|      0|    10452|      91.7|        9|                  f|        0|            0|
```

```
2270.7|         2270.7|        1256.14|       570.26|                0|     444.3|                  4.16|  01-07-2012
|         305.38|        NULL|        01-11-2012|                        0|                      NULL|          1|
INDIVIDUAL|           NULL|     NULL|                      NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          1|
|1069866|  1304956|     3000|       3000|         3000.0| 36 months|    9.91|      96.68|    B|       B1|
Target|   3 years|          RENT|     15000|    Source Verified|01-12-2011|         n|  Borrower added ...|       credi
t_card|       citicard fund|   606xx|        IL|12.56|          0|      01-07-2003|             2|                  NUL
L|                  NULL|      11|      0|     7323|      43.1|       11|                    f|        0|            0|34
78.981915|        3478.98|           3000|       478.98|                  0|         0|                       0|  01-01-2
015|         102.43|        NULL|        01-01-2016|                        0|                      NULL|          1|
INDIVIDUAL|           NULL|     NULL|                      NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1069057|  1303503|    10000|      10000|        10000.0| 36 months|   10.65|     325.74|    B|       B2|
SFMTA|   3 years|          RENT|    100000|    Source Verified|01-12-2011|         n|                NULL|
other|          Other Loan|   951xx|        CA| 7.06|          0|      01-05-1991|             2|                  NULL
|                  NULL|      14|      0|    11997|      55.5|       29|                    f|        0|            0|
7471.99|        7471.99|        5433.47|      1393.42|                  0|     645.1|                  6.3145|  01-10-201
3|         325.74|        NULL|        01-03-2014|                        0|                      NULL|          1|
INDIVIDUAL|           NULL|     NULL|                      NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          1|
|1069759|  1304871|     1000|       1000|         1000.0| 36 months|   16.29|      35.31|    D|       D1|Internal reven
ue ...|  < 1 year|          RENT|     28000|       Not Verified|01-12-2011|         n|                NULL|debt_consoli
dation|Debt Consolidatio...|   641xx|        MO|20.31|          0|      01-09-2007|             1|                  NUL
L|                  NULL|      11|      0|     6524|      81.5|       23|                    f|        0|            0|12
70.171106|        1270.17|           1000|       270.17|                  0|         0|                       0|  01-01-2
015|          36.32|        NULL|        01-01-2016|                        0|                      NULL|          1|
INDIVIDUAL|           NULL|     NULL|                      NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1065775|  1299699|    10000|      10000|        10000.0| 36 months|   15.27|     347.98|    C|       C4|   Chin's Rest
aurant|   4 years|          RENT|     42000|       Not Verified|01-12-2011|         n|                NULL|  home_impro
vement|                Home|   921xx|        CA| 18.6|          0|      01-10-1998|             2|                     6
1|                  NULL|      14|      0|    24043|      70.2|       28|                    f|        0|            0|12
519.26045|       12519.26|          10000|      2519.26|                  0|         0|                       0|  01-01-2
015|         370.46|        NULL|        01-04-2015|                        0|                      NULL|          1|
INDIVIDUAL|           NULL|     NULL|                      NULL|            0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|               NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1069971|  1304884|     3600|       3600|         3600.0| 36 months|    6.03|     109.57|    A|       A1|           Du
racell| 10+ years|      MORTGAGE|    110000|       Not Verified|01-12-2011|         n|  Borrower added ...|    major_pu
rchase|             Holiday|   067xx|        CT|10.52|          0|      01-08-1993|             0|                  NUL
L|                  NULL|      20|      0|    22836|        16|       42|                    f|        0|            0|
3785.02|        3785.02|           3600|       185.02|                  0|         0|                       0|  01-05-201
3|         583.45|        NULL|        01-05-2014|                        0|                      NULL|          1|
```

```
INDIVIDUAL|            NULL|     NULL|                    NULL|             0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1062474|  1294539|     6000|       6000|         6000.0| 36 months|   11.71|     198.46|    B|       B3|Connection Ins
pec...|    1 year|      MORTGAGE|     84000|           Verified|01-12-2011|         n|  Borrower added ...|           m
edical|             Medical|   890xx|        UT|18.44|           2|      01-10-2003|             0|
8|                NULL|       4|      0|        0|     37.73|       14|                   f|        0|           0|71
64.499852|         7164.5|           6000|       1149.5|                15|         0|                       0|  01-02-2
015|          16.98|        NULL|        01-07-2015|                        0|                        NULL|          1|
INDIVIDUAL|            NULL|     NULL|                    NULL|             0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
|1069742|  1304855|     9200|       9200|         9200.0| 36 months|    6.03|     280.01|    A|       A1|Network Interp
ret...|   6 years|          RENT|  77385.19|       Not Verified|01-12-2011|         n|                NULL|debt_consoli
dation|lowerratemeanseas...|   921xx|        CA| 9.86|           0|      01-01-2001|             0|                  NUL
L|                NULL|       8|      0|     7314|      23.1|       28|                   f|        0|           0|
9459.96|        9459.96|           9200|       259.96|                 0|         0|                       0|  01-07-201
2|         8061.1|        NULL|        01-07-2012|                        0|                        NULL|          1|
INDIVIDUAL|            NULL|     NULL|                    NULL|             0|        NULL|       NULL|       NULL|
NULL|       NULL|       NULL|              NULL|        NULL|   NULL|       NULL|       NULL|      NULL|    NULL|
NULL|  NULL|       NULL|        NULL|          0|
+-------+---------+---------+-----------+---------------+----------+--------+-----------+-----+---------+-------------
------+----------+-------------+----------+-------------------+----------+----------+--------------------+------------
------+--------------------+--------+----------+-----+-----------+----------------+-------------+--------------------
-+---------------------+--------+-------+---------+----------+---------+------------------+---------+-------------+--
---------+---------------+---------------+-------------+-------------------+----------+----------------------+---------
---+--------------+-----------+-----------------+-------------------------+-------------------------+-----------+
---------------+----------------+---------+-------------------------+-------------+------------+-----------+---------
--+----------+-----------+------+-----------+-----------------+-----------+-------+-----------+-----------+----------+------
--+---------------+------+-----------+-----------+-----------+
only showing top 20 rows
```

View General Stats

In [ ]:

```
# View data types and column names
df.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- member_id: integer (nullable = true)
 |-- loan_amnt: integer (nullable = true)
 |-- funded_amnt: integer (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: string (nullable = true)
 |-- verification_status: string (nullable = true)
 |-- issue_d: string (nullable = true)
 |-- pymnt_plan: 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: string (nullable = true)
 |-- delinq_2yrs: string (nullable = true)
 |-- earliest_cr_line: string (nullable = true)
 |-- inq_last_6mths: string (nullable = true)
 |-- mths_since_last_delinq: string (nullable = true)
 |-- mths_since_last_record: string (nullable = true)
 |-- open_acc: string (nullable = true)
 |-- pub_rec: string (nullable = true)
 |-- revol_bal: string (nullable = true)
 |-- revol_util: string (nullable = true)
 |-- total_acc: string (nullable = true)
 |-- initial_list_status: string (nullable = true)
 |-- out_prncp: string (nullable = true)
 |-- out_prncp_inv: string (nullable = true)
 |-- total_pymnt: string (nullable = true)
 |-- total_pymnt_inv: string (nullable = true)
 |-- total_rec_prncp: string (nullable = true)
 |-- total_rec_int: string (nullable = true)
 |-- total_rec_late_fee: string (nullable = true)
 |-- recoveries: string (nullable = true)
 |-- collection_recovery_fee: string (nullable = true)
 |-- last_pymnt_d: string (nullable = true)
 |-- last_pymnt_amnt: string (nullable = true)
 |-- next_pymnt_d: string (nullable = true)
 |-- last_credit_pull_d: string (nullable = true)
 |-- collections_12_mths_ex_med: string (nullable = true)
 |-- mths_since_last_major_derog: string (nullable = true)
 |-- policy_code: string (nullable = true)
 |-- application_type: string (nullable = true)
 |-- annual_inc_joint: string (nullable = true)
 |-- dti_joint: string (nullable = true)
```

```
 |-- verification_status_joint: string (nullable = true)
 |-- acc_now_delinq: string (nullable = true)
 |-- tot_coll_amt: string (nullable = true)
 |-- tot_cur_bal: string (nullable = true)
 |-- open_acc_6m: string (nullable = true)
 |-- open_il_6m: string (nullable = true)
 |-- open_il_12m: string (nullable = true)
 |-- open_il_24m: string (nullable = true)
 |-- mths_since_rcnt_il: string (nullable = true)
 |-- total_bal_il: string (nullable = true)
 |-- il_util: string (nullable = true)
 |-- open_rv_12m: string (nullable = true)
 |-- open_rv_24m: string (nullable = true)
 |-- max_bal_bc: string (nullable = true)
 |-- all_util: string (nullable = true)
 |-- total_rev_hi_lim: string (nullable = true)
 |-- inq_fi: string (nullable = true)
 |-- total_cu_tl: string (nullable = true)
 |-- inq_last_12m: integer (nullable = true)
 |-- default_ind: integer (nullable = true)
```

In [ ]:

```
df.describe().show()
```

```
+-------+--------------------+--------------------+------------------+------------------+------------------+------------------+----------+
------------------+-----------------+------+---------+--------------------+----------+--------------+------------------
+-------------------+--------------+----------+--------------------+--------------------+--------------------+--------
-----------+--------------------+-----------------+--------------------+------------------+------------------+---------
------------+--------------------+-----------------+--------------------+--------------------+--------------------+------
--------------------+--------------------+-----------------+--------------------+--------------------+--------------------
+--------------------+------------------+--------------------+--------------------+--------------------+--------------------+--
----+------------------+------------------+-----------------+--------------------+--------------------+--
----------------+-----------------+-----------------+-----------------+--------------------+--------------------+-
----------------+------------------+-------------------+------------------+------------------+------------------+-------
----------+-----------------+-----------------+-----------------+------------------+------------------+--------------
---+------------------+------------------+-----------------+------------------+--------------------+
|summary|                  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|verificat
ion_status|        issue_d|pymnt_plan|               desc|             purpose|               title|           zip_co
de|          addr_state|               dti|       delinq_2yrs|  earliest_cr_line|    inq_last_6mths|mths_since_last_del
inq|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|   las
t_pymnt_amnt|      next_pymnt_d|last_credit_pull_d|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_co
ll_amt|       tot_cur_bal|       open_acc_6m|        open_il_6m|       open_il_12m|      open_il_24m|mths_since_rcnt_il
```

```
|     total_bal_il|          il_util|       open_rv_12m|       open_rv_24m|       max_bal_bc|          all_util|  total
_rev_hi_lim|           inq_fi|       total_cu_tl|      inq_last_12m|        default_ind|
+-------+------------------+------------------+------------------+------------------+-----------------+----------+
-----------------+-----------------+------+--------+------------------+----------+-------------+----------------
+------------------+-------------+----------+------------------+------------------+-------------------+--------
-----------+------------------+-----------------+-----------------+-----------------+-----------------+--------
-----------+------------------+-----------------+------------------+-----------------+-----------------+
------------------+------------------+-----------------+------------------+-----------------+-----------------
+------------------+-----------------+------------------+-----------------+----------------------+-------------
----+-----------------+-----------------+-----------------+----------------------+----------------------+--
----------------+----------------+-----------------+-----------------+----------------------+------------------+-
----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-------
----------+----------------+-----------------+-----------------+-----------------+-----------------+-------------
---+-----------------+-----------------+-----------------+-----------------+------------------+
|  count|             855969|             855969|             855969|             855969|            855969|    855969|
855969|            855969|855969|   855969|              806530|    855969|        855969|            855969|
855969|         855969|    855969|              121814|               855968|             855936|               855968|
855968|            855968|            855968|            855968|            855968|                416300|
131356|            855897|             855912|             855922|             855494|               855952|
855955|           855954|             855948|            855957|              855959|             855960|             8
55966|             855966|            855967|                 855964|             847100|             855966|
603218|            855867|                  855870|                       213332|             855897|            855912|
626|               587|                    551|              855826|             788771|             788745|
13369|             13353|             13331|            13322|              12959|             13314|             11632|
13299|             13305|            13301|             13294|             788658|              13293|              13297|
13289|             855746|
|   mean|3.2240726286145877E7|3.4762690042003855E7|14745.571334943204|14732.378304588134|14700.061225999756|      NULL|
13.192319605055827| 436.2380718344112|  NULL|     NULL|1.2810941556818182E7|      NULL|          NULL|75071.22927176002
|            38000.0|           NULL|      NULL|                NULL|             384.0|           Infinity|
0.0|               NULL|18.123810127573403|0.3123480678154647|26.896086956521735|0.6822168381595152|    34.14263405890
2544|     70.42372760102394| 11.54296815097585|  0.1960765625712012|   16908.69626299838|   55.91715695820985|  25.9682
00584869795|  2263.0536559139778| 8285.89549835517|   8282.436991670364|7650.628777464625|   7620.439186803648|   5849.
749878239475|1756.4741183920426|  1.7748607324743964| 48.21955402103537|      5.620354049139619|1976.5201959016651|222
5.9697361985486|2717.9857799171273|3367.8102573918713|       0.38861753160172524|          45.53334005049339|1.19557966
74413361|720.6995652173914|86393.42756045342|255.30688804554083|        2319.298032786885|0.10623780751160922| 225.4549
463071762|139753.06323234446| 18.59975509286998|3.5713591723517504|1.5689154273696861|2.734639128802103| 22.10486448922
863|36465.59811198797|71.37272508384228|1.4044558923065353|2.9427202345159347|5838.685826060789|61.009643421349395|3216
3.519162322784|1.8130173062452972| 1.523315282791817|1.8418240650161788|0.05426376518265934|
| stddev|2.2719691565777678E7| 2.399417679681356E7| 8425.340005005204| 8419.471653319737| 8425.805478470169|      NULL|
4.368365181701446|243.72687616798578|  NULL|     NULL|  8.34575692812245E7|      NULL|          NULL|64264.49486103645|
NULL|           NULL|      NULL|                NULL|   543.0580079512685|                NaN|               NULL|
NULL|17.425265176637534|0.8637496886299738|209.15285346366267|1.3034984573776172|    21.871669441275337|     27.8635882
8124207|5.3120462644240565|  0.6343819932768587|  22224.042643877146|   179.7016450953392|   163.6776277696635|   7051.
063987047671|8462.087431679473|    8458.62393772719|7909.012728877155|    7885.00474480922|   6676.542304929754| 2093.2
89304829823|  154.32649175739448|433.81372107990103|     116.56184154389102| 4954.840516380205| 4865.299583611445| 534
9.237639779675| 6331.137195930885|         74.56147279197249|          215.3908396117096| 50.03794040401378|2933.559501
651738|60512.88057230068|1608.7839691240242|        4688.120511662365|  31.29304993111533|10488.878812417901|153937.306
```

```
59434243|1911.5780476942048| 69.40644079814113|60.390178110053284| 68.7627527866157| 109.9153258487779|42485.0576086488
3|23.17729877069355| 4.432960390363493| 2.595224748408899|5108.735931018706|20.038139706725396| 37699.64159437814| 99.7
5735040533151|2.6970485485871754| 2.974979879227414|0.22653756629105157|
|    min|               54734|               70699|               500|               500|               0.0| 36 months|
5.32|            15.69|     A|       A1| \tAdv Mtr Proj F...| reactors"|       2 years|                0|
38000|     01-01-2008|01-10-2015|\t Loan for purch...|      After gradu...|       \tcredit_card| ""set it and for...| A
merican women u...|       Karen     "|          BEST BUY|           Mike  "|               809|   Capital One ($27...|
I am currently ...|           Carlos"| I plan on elimin...| I'm sure)!!  Ple...| #450216.<br/> 55...| and the work we
...| I am working on ...|    I mean.<br/>"| and I pay her tu...|     and opinions| extremely rare t...| I need to hit t
h...|             of $2| and continue to ...|                  0|                      0|                 0|
0|                 0|                 0|                    0|                          0|                 0|
0|                0|                0|                     0|                   0|                0|
0|                 0|                 0|                0|                0|                 0|                 0|
0|                 0|                 0|               0|                0|                 0|                 0|
0|                -4|                 0|
|    max|            68616867|            73519693|             35000|             35000|          35000.0| 60 months|
28.99|          1445.46|     G|       G5|â€‹License Compli...|       n/a|         RENT|         MORTGAGE|           V
erified|Source Verified|         y|~20K total in med...|             wedding|Ã®Â□Â–Ã®Â□Â–MY FI...|      small_business|
wedding|           wedding|    small_business|    small_business|    small_business|      stupid decisions|        smal
l_business|  home_improvement|    home_improvement|  debt_consolidation|              moving|      small_business|
w|   small_business|               other|                f|savy debt consoli...|                  f|                f
|                  f|    small_business|   xperienceddebtorq...|                f|                f|
f|                 f|                          f|                          99|                f|               f|
f|                 f|                         f|        INDIVIDUAL|       INDIVIDUAL|                f|       INDIVI
DUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|
INDIVIDUAL|                 9|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVIDUAL|       INDIVI
DUAL|               32|                1|
+-------+------------------+-----------------+-----------------+-----------------+-----------------+-----------------+----------+
-----------------+-----------------+------+---------+-------------------+----------+-------------+-----------------+--------
+-----------------+-------------+----------+-------------------+-------------------+-------------------+--------
-----------+-------------------+-----------------+-----------------+-----------------+-----------------+---------
-----------+-------------------+-----------------+-------------------+-------------------+-------------------+
-------------------+-----------------+---------------+-----------------+----------------+-------------------
+-------------------+----------------+-----------------+-----------------+----------------------+-----------------
----+-----------------+-----------------+-----------------+----------------------+----------------------+--
---------------+----------------+----------------+-----------------+----------------------+-------------------+-
---------------+----------------+-----------------+-----------------+-----------------+----------------+-------
----------+-----------------+-----------------+-----------------+-----------------+-----------------+--------------
---+-----------------+-----------------+-----------------+-----------------+-------------------+
```

In [ ]:

```
# Get the number of rows
num_rows = df.count()

# Get the number of columns
num_columns = len(df.columns)
```

```
print(f"Number of rows: {num_rows}")
print(f"Number of columns: {num_columns}")
```

```
Number of rows: 855969
Number of columns: 73
```

Remove and check nulls before finding correlation

Remove columns with more than 50% null values.

In [ ]:

```
from pyspark.sql.functions import col, count, when, isnan

total_rows = df.count()

missing_data = []

for c in df.columns:
  null_count = df.select(count(when(col(c).isNull()|isnan(c), c)).alias('Count')).collect()[0]['Count']
  percent = (null_count/total_rows)*100
  missing_data.append((c, null_count, percent))

df_null = spark.createDataFrame(missing_data, ["Column Name", "Count", "Percent"])

columns_to_drop = df_null.filter(col("Percent")>50).select("Column Name").rdd.flatMap(lambda x:x).collect()

df = df.drop(*columns_to_drop)

df.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- member_id: integer (nullable = true)
 |-- loan_amnt: integer (nullable = true)
 |-- funded_amnt: integer (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: string (nullable = true)
|-- verification_status: string (nullable = true)
|-- issue_d: string (nullable = true)
|-- pymnt_plan: string (nullable = true)
|-- purpose: string (nullable = true)
|-- title: string (nullable = true)
|-- zip_code: string (nullable = true)
|-- addr_state: string (nullable = true)
|-- dti: string (nullable = true)
|-- delinq_2yrs: string (nullable = true)
|-- earliest_cr_line: string (nullable = true)
|-- inq_last_6mths: string (nullable = true)
|-- open_acc: string (nullable = true)
|-- pub_rec: string (nullable = true)
|-- revol_bal: string (nullable = true)
|-- revol_util: string (nullable = true)
|-- total_acc: string (nullable = true)
|-- initial_list_status: string (nullable = true)
|-- out_prncp: string (nullable = true)
|-- out_prncp_inv: string (nullable = true)
|-- total_pymnt: string (nullable = true)
|-- total_pymnt_inv: string (nullable = true)
|-- total_rec_prncp: string (nullable = true)
|-- total_rec_int: string (nullable = true)
|-- total_rec_late_fee: string (nullable = true)
|-- recoveries: string (nullable = true)
|-- collection_recovery_fee: string (nullable = true)
|-- last_pymnt_d: string (nullable = true)
|-- last_pymnt_amnt: string (nullable = true)
|-- next_pymnt_d: string (nullable = true)
|-- last_credit_pull_d: string (nullable = true)
|-- collections_12_mths_ex_med: string (nullable = true)
|-- policy_code: string (nullable = true)
|-- application_type: string (nullable = true)
|-- acc_now_delinq: string (nullable = true)
|-- tot_coll_amt: string (nullable = true)
|-- tot_cur_bal: string (nullable = true)
|-- total_rev_hi_lim: string (nullable = true)
|-- default_ind: integer (nullable = true)
```

In [ ]:

```
total_rows = df.count()

missing_data = []

for c in df.columns:
```

```
  null_count = df.select(count(when(col(c).isNull()|isnan(c), c)).alias('Count')).collect()[0]['Count']
  percent = (null_count/total_rows)*100
  missing_data.append((c, null_count, percent))

df_null = spark.createDataFrame(missing_data, ["Column Name", "Count", "Percent"])

df_null.show(df_null.count(), truncate=False)
```

```
+--------------------------+------+--------------------+
|Column Name               |Count |Percent             |
+--------------------------+------+--------------------+
|id                        |0     |0.0                 |
|member_id                 |0     |0.0                 |
|loan_amnt                 |0     |0.0                 |
|funded_amnt               |0     |0.0                 |
|funded_amnt_inv           |0     |0.0                 |
|term                      |0     |0.0                 |
|int_rate                  |0     |0.0                 |
|installment               |0     |0.0                 |
|grade                     |0     |0.0                 |
|sub_grade                 |0     |0.0                 |
|emp_title                 |49439 |5.7757932822333515  |
|emp_length                |0     |0.0                 |
|home_ownership            |0     |0.0                 |
|annual_inc                |0     |0.0                 |
|verification_status       |0     |0.0                 |
|issue_d                   |0     |0.0                 |
|pymnt_plan                |0     |0.0                 |
|purpose                   |1     |1.1682666077860296E-4|
|title                     |33    |0.003855279805693898 |
|zip_code                  |1     |1.1682666077860296E-4|
|addr_state                |1     |1.1682666077860296E-4|
|dti                       |1     |1.1682666077860296E-4|
|delinq_2yrs               |1     |1.1682666077860296E-4|
|earliest_cr_line          |1     |1.1682666077860296E-4|
|inq_last_6mths            |1     |1.1682666077860296E-4|
|open_acc                  |72    |0.008411519576059415 |
|pub_rec                   |57    |0.006659119664380368 |
|revol_bal                 |47    |0.005490853056594339 |
|revol_util                |475   |0.055492663869836416 |
|total_acc                 |17    |0.0019860532332362504|
|initial_list_status       |14    |0.0016355732509004417|
|out_prncp                 |15    |0.0017523999116790445|
|out_prncp_inv             |21    |0.002453359876350662 |
|total_pymnt               |12    |0.0014019199293432356|
|total_pymnt_inv           |10    |0.0011682666077860297|
|total_rec_prncp           |9     |0.0010514399470074268|
```

```
|total_rec_int              |3     |3.504799823358089E-4 |
|total_rec_late_fee         |3     |3.504799823358089E-4 |
|recoveries                 |2     |2.3365332155720592E-4|
|collection_recovery_fee    |5     |5.841333038930148E-4 |
|last_pymnt_d               |8869  |1.0361356544454297   |
|last_pymnt_amnt            |3     |3.504799823358089E-4 |
|next_pymnt_d               |252751|29.52805533845268    |
|last_credit_pull_d         |102   |0.011916319399417502 |
|collections_12_mths_ex_med|99    |0.011565839417081693 |
|policy_code                |72    |0.008411519576059415 |
|application_type           |57    |0.006659119664380368 |
|acc_now_delinq             |143   |0.016706212491340224 |
|tot_coll_amt               |67198 |7.850517951000562    |
|tot_cur_bal                |67224 |7.853555444180806    |
|total_rev_hi_lim           |67311 |7.863719363668544    |
|default_ind                |223   |0.02605234535362846  |
+---------------------------+------+---------------------+
```

Convert numerical fields to categorical

In [ ]:

```
from pyspark.sql.types import IntegerType, DoubleType, StringType

# Correct numerical fields that are incorrectly typed as strings
numerical_fields = ['annual_inc', 'dti', 'delinq_2yrs', 'inq_last_6mths', 'open_acc',
                    'pub_rec', 'revol_bal', 'revol_util', 'total_acc', '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_amnt', 'collections_12_mths_ex_med', 'tot_coll_amt', 'tot_cur_bal',
                    'total_rev_hi_lim']
for field in numerical_fields:
    df = df.withColumn(field, col(field).cast(DoubleType()))
```

In [ ]:

```
df.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- member_id: integer (nullable = true)
 |-- loan_amnt: integer (nullable = true)
 |-- funded_amnt: integer (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)
|-- pymnt_plan: 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)
|-- inq_last_6mths: 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)
|-- collections_12_mths_ex_med: double (nullable = true)
|-- policy_code: string (nullable = true)
|-- application_type: string (nullable = true)
|-- acc_now_delinq: string (nullable = true)
|-- tot_coll_amt: double (nullable = true)
|-- tot_cur_bal: double (nullable = true)
|-- total_rev_hi_lim: double (nullable = true)
|-- default_ind: integer (nullable = true)
```

In [ ]:
```
df.show(5)
```

```
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+-------------
------+---------+-------------+---------+------------------+---------+---------+-------------+----------------
--+--------+----------+-----+----------+---------------+-------------+--------+-------+---------+----------+--------
-+------------------+---------+-------------+-----------+---------------+---------------+------------+---------------
---+---------+--------------------+-------------+-------------+-----------+----------------+-----------+
|     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|pymnt_plan|       purpose|               tit
le|zip_code|addr_state|  dti|delinq_2yrs|earliest_cr_line|inq_last_6mths|open_acc|pub_rec|revol_bal|revol_util|total_ac
c|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|collections_12_mths
_ex_med|policy_code|application_type|acc_now_delinq|tot_coll_amt|tot_cur_bal|total_rev_hi_lim|default_ind|
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+-------------
------+---------+-------------+---------+------------------+---------+---------+-------------+----------------
--+--------+----------+-----+----------+---------------+-------------+--------+-------+---------+----------+--------
-+------------------+---------+-------------+-----------+---------------+---------------+------------+---------------
---+---------+--------------------+-------------+-------------+-----------+----------------+-----------+
|1077501|  1296599|     5000|       5000|         4975.0| 36 months|   10.65|     162.87|    B|       B2|
NULL| 10+ years|          RENT|   24000.0|           Verified|01-12-2011|         n|   credit_card|            Computer
|   860xx|        AZ|27.65|        0.0|      01-01-1985|          1.0|     3.0|    0.0|  13648.0|      83.7|      9.0|
f|      0.0|          0.0|5861.071414|        5831.78|        5000.0|       861.07|               0.0|       0.0|
0.0|  01-01-2015|         171.62|        NULL|        01-01-2016|                        0.0|          1|      INDIVIDUA
L|             0|        NULL|       NULL|            NULL|          0|
|1077430|  1314167|     2500|       2500|         2500.0| 60 months|   15.27|      59.83|    C|       C4|
Ryder|  < 1 year|          RENT|   30000.0|    Source Verified|01-12-2011|         n|           car|                bik
e|   309xx|        GA|  1.0|        0.0|      01-04-1999|          5.0|     3.0|    0.0|   1687.0|       9.4|      4.0
|                 f|      0.0|          0.0|    1008.71|        1008.71|         456.46|       435.17|
0.0|    117.08|                    1.11|  01-04-2013|        119.66|        NULL|        01-09-2013|
0.0|          1|      INDIVIDUAL|              0|        NULL|       NULL|            NULL|          1|
|1077175|  1313524|     2400|       2400|         2400.0| 36 months|   15.96|      84.33|    C|       C5|
NULL| 10+ years|          RENT|   12252.0|       Not Verified|01-12-2011|         n|small_business|real estate business
|   606xx|        IL| 8.72|        0.0|      01-11-2001|          2.0|     2.0|    0.0|   2956.0|      98.5|     10.0|
f|      0.0|          0.0|3003.653644|        3003.65|        2400.0|       603.65|               0.0|       0.0|
0.0|  01-06-2014|         649.91|        NULL|        01-01-2016|                        0.0|          1|      INDIVIDUA
L|             0|        NULL|       NULL|            NULL|          0|
|1076863|  1277178|    10000|      10000|        10000.0| 36 months|   13.49|     339.31|    C|       C1| AIR RESOURCES
BOARD| 10+ years|          RENT|   49200.0|    Source Verified|01-12-2011|         n|         other|             persone
l|   917xx|        CA| 20.0|        0.0|      01-02-1996|          1.0|    10.0|    0.0|   5598.0|      21.0|     37.0
|                 f|      0.0|          0.0|12226.30221|        12226.3|        10000.0|      2209.33|             16.
97|       0.0|                     0.0|  01-01-2015|         357.48|        NULL|        01-01-2015|
0.0|          1|      INDIVIDUAL|              0|        NULL|       NULL|            NULL|          0|
|1075358|  1311748|     3000|       3000|         3000.0| 60 months|   12.69|      67.79|    B|       B5|University Med
ica...|    1 year|          RENT|   80000.0|    Source Verified|01-12-2011|         n|         other|             Person
```

```
al|   972xx|        OR|17.94|        0.0|      01-01-1996|           0.0|    15.0|    0.0|  27783.0|      53.9|     38.
0|                 f|    766.9|        766.9|    3242.17|        3242.17|         2233.1|      1009.07|
0.0|       0.0|                   0.0|  01-01-2016|          67.79|  01-02-2016|        01-01-2016|
0.0|          1|      INDIVIDUAL|             0|        NULL|       NULL|            NULL|          0|
+-------+---------+---------+----------+---------------+---------+--------+----------+-----+---------+-------------
------+---------+-------------+---------+-----------------+---------+---------+-------------+-----------------
--+--------+----------+-----+----------+---------------+-------------+--------+-------+---------+---------+--------
-+------------------+---------+-------------+----------+---------------+---------------+------------+--------------
---+----------+----------------------+------------+--------------+------------+-----------------+-----------------
-------+----------+---------------+-------------+------------+----------+---------------+----------+
only showing top 5 rows
```

Apply Imputation

Apply median imputation to the columns which seem necessary for the prediction analysis.

In [ ]:

```
imputer = Imputer(
    inputCols=["tot_coll_amt", "tot_cur_bal", "total_rev_hi_lim"],
    outputCols=["tot_coll_amt", "tot_cur_bal", "total_rev_hi_lim"],
    strategy="median"
)

# Apply the imputer
df = imputer.fit(df).transform(df)
```

In [ ]:

```
total_rows = df.count()

missing_data = []

for c in df.columns:
  null_count = df.select(count(when(col(c).isNull()|isnan(c), c)).alias('Count')).collect()[0]['Count']
  percent = (null_count/total_rows)*100
  missing_data.append((c, null_count, percent))

df_null = spark.createDataFrame(missing_data, ["Column Name", "Count", "Percent"])

df_null.show(df_null.count(), truncate=False)
```

```
+------------------------+------+--------------------+
|Column Name             |Count |Percent             |
+------------------------+------+--------------------+
```

```
|id                          |0     |0.0                  |
|member_id                   |0     |0.0                  |
|loan_amnt                   |0     |0.0                  |
|funded_amnt                 |0     |0.0                  |
|funded_amnt_inv             |0     |0.0                  |
|term                        |0     |0.0                  |
|int_rate                    |0     |0.0                  |
|installment                 |0     |0.0                  |
|grade                       |0     |0.0                  |
|sub_grade                   |0     |0.0                  |
|emp_title                   |49439 |5.7757932822333515   |
|emp_length                  |0     |0.0                  |
|home_ownership              |0     |0.0                  |
|annual_inc                  |1     |1.1682666077860296E-4|
|verification_status         |0     |0.0                  |
|issue_d                     |0     |0.0                  |
|pymnt_plan                  |0     |0.0                  |
|purpose                     |1     |1.1682666077860296E-4|
|title                       |33    |0.003855279805693898 |
|zip_code                    |1     |1.1682666077860296E-4|
|addr_state                  |1     |1.1682666077860296E-4|
|dti                         |226   |0.02640282533596427  |
|delinq_2yrs                 |176   |0.020561492297034124 |
|earliest_cr_line            |1     |1.1682666077860296E-4|
|inq_last_6mths              |151   |0.017640825777569046 |
|open_acc                    |150   |0.017523999116790444 |
|pub_rec                     |120   |0.014019199293432356 |
|revol_bal                   |93    |0.010864879452410076 |
|revol_util                  |512   |0.059815250318644715 |
|total_acc                   |52    |0.0060749863604873545|
|initial_list_status         |14    |0.0016355732509004417|
|out_prncp                   |84    |0.009813439505402649 |
|out_prncp_inv               |82    |0.009579786183845444 |
|total_pymnt                 |59    |0.006892772985937575 |
|total_pymnt_inv             |44    |0.005140373074258531 |
|total_rec_prncp             |29    |0.0033879731625794858|
|total_rec_int               |16    |0.0018692265724576473|
|total_rec_late_fee          |22    |0.0025701865371292655|
|recoveries                  |18    |0.0021028798940148537|
|collection_recovery_fee     |18    |0.0021028798940148537|
|last_pymnt_d                |8869  |1.0361356544454297   |
|last_pymnt_amnt             |63    |0.007360079629051986 |
|next_pymnt_d                |252751|29.52805533845268    |
|last_credit_pull_d          |102   |0.011916319399417502 |
|collections_12_mths_ex_med  |173   |0.020211012314698313 |
|policy_code                 |72    |0.008411519576059415 |
|application_type            |57    |0.006659119664380368 |
```

```
|acc_now_delinq            |143   |0.016706212491340224 |
|tot_coll_amt              |0     |0.0                  |
|tot_cur_bal               |0     |0.0                  |
|total_rev_hi_lim          |0     |0.0                  |
|default_ind               |223   |0.02605234535362846  |
+--------------------------+------+---------------------+
```

In [ ]:

```
df.show(5)
```

```
+-------+---------+---------+-----------+---------------+----------+--------+-----------+-----+---------+-------------
------+----------+--------------+----------+------------------+----------+----------+--------------+-----------------
--+--------+----------+-----+----------+---------------+-------------+--------+-------+--------+----------+--------
-+------------------+---------+------------+-----------+---------------+--------------+-----------+---------------
---+----------+----------------------+-----------+-------------+-------------+-----------+----------------+-----------+
|     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|pymnt_plan|       purpose|               tit
le|zip_code|addr_state|  dti|delinq_2yrs|earliest_cr_line|inq_last_6mths|open_acc|pub_rec|revol_bal|revol_util|total_ac
c|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|collections_12_mths
_ex_med|policy_code|application_type|acc_now_delinq|tot_coll_amt|tot_cur_bal|total_rev_hi_lim|default_ind|
+-------+---------+---------+-----------+---------------+----------+--------+-----------+-----+---------+-------------
------+----------+--------------+----------+------------------+----------+----------+--------------+-----------------
--+--------+----------+-----+----------+---------------+-------------+--------+-------+--------+----------+--------
-+------------------+---------+------------+-----------+---------------+--------------+-----------+---------------
---+----------+----------------------+-----------+-------------+-------------+-----------+----------------+-----------+
|1077501|  1296599|     5000|       5000|         4975.0| 36 months|   10.65|     162.87|    B|       B2|
NULL| 10+ years|          RENT|   24000.0|           Verified|01-12-2011|         n|   credit_card|            Computer
|   860xx|        AZ|27.65|        0.0|      01-01-1985|           1.0|     3.0|    0.0|  13648.0|      83.7|      9.0|
f|      0.0|           0.0|5861.071414|        5831.78|         5000.0|       861.07|               0.0|       0.0|
0.0|  01-01-2015|         171.62|        NULL|        01-01-2016|                        0.0|          1|      INDIVIDUA
L|             0|         0.0|    80935.0|         23800.0|          0|
|1077430|  1314167|     2500|       2500|         2500.0| 60 months|   15.27|      59.83|    C|       C4|
Ryder|  < 1 year|          RENT|   30000.0|    Source Verified|01-12-2011|         n|           car|                bik
e|   309xx|        GA|  1.0|        0.0|      01-04-1999|           5.0|     3.0|    0.0|   1687.0|       9.4|      4.0
|                 f|      0.0|           0.0|    1008.71|        1008.71|         456.46|       435.17|
0.0|    117.08|                   1.11|  01-04-2013|         119.66|        NULL|        01-09-2013|
0.0|          1|      INDIVIDUAL|             0|         0.0|    80935.0|         23800.0|          1|
|1077175|  1313524|     2400|       2400|         2400.0| 36 months|   15.96|      84.33|    C|       C5|
NULL| 10+ years|          RENT|   12252.0|       Not Verified|01-12-2011|         n|small_business|real estate business
|   606xx|        IL| 8.72|        0.0|      01-11-2001|           2.0|     2.0|    0.0|   2956.0|      98.5|     10.0|
f|      0.0|           0.0|3003.653644|        3003.65|         2400.0|       603.65|               0.0|       0.0|
0.0|  01-06-2014|         649.91|        NULL|        01-01-2016|                        0.0|          1|      INDIVIDUA
L|             0|         0.0|    80935.0|         23800.0|          0|
```

```
|1076863|  1277178|    10000|      10000|        10000.0| 36 months|   13.49|     339.31|    C|       C1| AIR RESOURCES
BOARD| 10+ years|          RENT|   49200.0|    Source Verified|01-12-2011|         n|         other|            persone
l|   917xx|        CA| 20.0|        0.0|      01-02-1996|           1.0|    10.0|    0.0|   5598.0|      21.0|     37.0
|                  f|      0.0|          0.0|12226.30221|        12226.3|        10000.0|     2209.33|             16.
97|       0.0|                     0.0|  01-01-2015|         357.48|        NULL|        01-01-2015|
0.0|          1|      INDIVIDUAL|              0|         0.0|    80935.0|         23800.0|          0|
|1075358|  1311748|     3000|       3000|         3000.0| 60 months|   12.69|      67.79|    B|       B5|University Med
ica...|    1 year|          RENT|   80000.0|    Source Verified|01-12-2011|          n|         other|             Person
al|   972xx|        OR|17.94|        0.0|      01-01-1996|           0.0|    15.0|    0.0|  27783.0|      53.9|     38.
0|                  f|    766.9|        766.9|    3242.17|        3242.17|         2233.1|      1009.07|
0.0|       0.0|                     0.0|  01-01-2016|          67.79|  01-02-2016|        01-01-2016|
0.0|          1|      INDIVIDUAL|              0|         0.0|    80935.0|         23800.0|          0|
+-------+---------+---------+-----------+--------------+----------+--------+----------+-----+---------+--------------
------+----------+--------------+----------+------------------+----------+----------+--------------+-----------------
--+--------+----------+-----+-----------+-----------------+--------------+--------+-------+---------+-----------+--------
-+-------------------+---------+-------------+-----------+---------------+----------------+-------------+---------------
---+----------+----------------------+------------+---------------+------------+------------------+-------------------
-------+-----------+----------------+-------------+------------+-----------+----------------+-----------+
only showing top 5 rows
```

Drop rows by checking null values and threshold

In [ ]:

```
df = df.drop("next_pymnt_d")
df = df.filter(col('default_ind').isNotNull())
df = df.na.drop()
```

In [ ]:

```
# Define the dominance threshold
dominance_threshold = 99.0

# List to hold columns to drop
columns_to_drop = []

# Analyze each column in the DataFrame
for col_name in df.columns:
    # Get the total number of records
    total_count = df.count()

    # Find the most frequent value and its count for the current column
    top_value_data = df.groupBy(col_name).agg(count(col_name).alias('count')).orderBy('count', ascending=False).first()

    # Calculate the percentage of the most frequent value
```

```
    if top_value_data and total_count > 0:  # Ensure there is data and avoid division by zero
        top_value_percentage = (top_value_data['count'] / total_count) * 100
        print(f"{col_name} - Top value percentage: {top_value_percentage}%")

        # Check if the top value percentage exceeds the dominance threshold
        if top_value_percentage > dominance_threshold:
            columns_to_drop.append(col_name)

# Drop columns where the top value percentage exceeds the threshold
if columns_to_drop:
    df = df.drop(*columns_to_drop)
    print(f"Dropped columns: {columns_to_drop}")
else:
    print("No columns exceeded the dominance threshold; no columns dropped.")

# Optionally, print the schema of the DataFrame to confirm the drops
df.printSchema()
```

```
id - Top value percentage: 0.00012538807609551562%
member_id - Top value percentage: 0.00012538807609551562%
loan_amnt - Top value percentage: 6.969696209769236%
funded_amnt - Top value percentage: 6.9586620590728305%
funded_amnt_inv - Top value percentage: 6.333978663964972%
term - Top value percentage: 69.47414748647063%
int_rate - Top value percentage: 4.027966556492344%
installment - Top value percentage: 0.29754590457465857%
grade - Top value percentage: 28.973422743390792%
sub_grade - Top value percentage: 6.426765840275653%
emp_title - Top value percentage: 1.6052181501747909%
emp_length - Top value percentage: 34.653377202441554%
home_ownership - Top value percentage: 50.34694880655629%
annual_inc - Top value percentage: 3.8810117313083996%
verification_status - Top value percentage: 38.147190554767%
issue_d - Top value percentage: 5.6803306232790485%
pymnt_plan - Top value percentage: 99.99937305961953%
purpose - Top value percentage: 59.45877490834132%
title - Top value percentage: 46.69978583716603%
zip_code - Top value percentage: 1.1150761607174204%
addr_state - Top value percentage: 14.676423530827911%
dti - Top value percentage: 0.07849293563579278%
delinq_2yrs - Top value percentage: 80.75092411012082%
earliest_cr_line - Top value percentage: 0.7673750257045556%
inq_last_6mths - Top value percentage: 56.38739398438166%
open_acc - Top value percentage: 9.027063762344456%
pub_rec - Top value percentage: 85.27893831408208%
revol_bal - Top value percentage: 0.3260089978483406%
```

```
revol_util - Top value percentage: 0.38556833399371054%
total_acc - Top value percentage: 3.6450313720966387%
initial_list_status - Top value percentage: 51.788911681654724%
out_prncp - Top value percentage: 29.861922650603617%
out_prncp_inv - Top value percentage: 29.861922650603617%
total_pymnt - Top value percentage: 0.050155230438206244%
total_pymnt_inv - Top value percentage: 0.06695723263500535%
total_rec_prncp - Top value percentage: 1.7392580035208973%
total_rec_int - Top value percentage: 0.058681619612701304%
total_rec_late_fee - Top value percentage: 98.83301317577904%
recoveries - Top value percentage: 97.22817118983254%
collection_recovery_fee - Top value percentage: 97.35945250550454%
last_pymnt_d - Top value percentage: 55.135268656491846%
last_pymnt_amnt - Top value percentage: 0.23961661341853036%
last_credit_pull_d - Top value percentage: 81.9314528465601%
collections_12_mths_ex_med - Top value percentage: 98.7026095766397%
policy_code - Top value percentage: 100.0%
application_type - Top value percentage: 99.95962503949724%
acc_now_delinq - Top value percentage: 99.5384464918924%
tot_coll_amt - Top value percentage: 87.00603367422171%
tot_cur_bal - Top value percentage: 7.8658447896238854%
total_rev_hi_lim - Top value percentage: 8.051544530321344%
default_ind - Top value percentage: 94.68642949930035%
Dropped columns: ['pymnt_plan', 'policy_code', 'application_type', 'acc_now_delinq']
root
 |-- id: integer (nullable = true)
 |-- member_id: integer (nullable = true)
 |-- loan_amnt: integer (nullable = true)
 |-- funded_amnt: integer (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)
 |-- 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)
 |-- inq_last_6mths: 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)
 |-- last_credit_pull_d: string (nullable = true)
 |-- collections_12_mths_ex_med: double (nullable = true)
 |-- tot_coll_amt: double (nullable = true)
 |-- tot_cur_bal: double (nullable = true)
 |-- total_rev_hi_lim: double (nullable = true)
 |-- default_ind: integer (nullable = true)
```

In [ ]:

```
df.show(5)
```

```
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+--------------
------+----------+--------------+----------+--------------------+----------+--------------------+----------------------+----
----+----------+-----+-----------+----------------+--------------+--------+-------+---------+----------+---------+-----
--------------+---------+-------------+-----------+---------------+---------------+-------------+------------------+---
-------+----------------------+------------+--------------+------------------+--------------------------+------------
+-----------+----------------+-----------+
|     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|             purpose|                 title|zip_
code|addr_state|  dti|delinq_2yrs|earliest_cr_line|inq_last_6mths|open_acc|pub_rec|revol_bal|revol_util|total_acc|initi
al_list_status|out_prncp|out_prncp_inv|total_pymnt|total_pymnt_inv|total_rec_prncp|total_rec_int|total_rec_late_fee|rec
overies|collection_recovery_fee|last_pymnt_d|last_pymnt_amnt|last_credit_pull_d|collections_12_mths_ex_med|tot_coll_amt
|tot_cur_bal|total_rev_hi_lim|default_ind|
+-------+---------+---------+-----------+---------------+---------+--------+-----------+-----+---------+--------------
------+----------+--------------+----------+--------------------+----------+--------------------+----------------------+----
----+----------+-----+-----------+----------------+--------------+--------+-------+---------+----------+---------+-----
--------------+---------+-------------+-----------+---------------+---------------+-------------+------------------+---
-------+----------------------+------------+--------------+------------------+--------------------------+------------
```

```
+-----------+---------------+-----------+
|1077430|  1314167|     2500|       2500|         2500.0| 60 months|   15.27|      59.83|    C|       C4|
Ryder|  < 1 year|          RENT|   30000.0|    Source Verified|01-12-2011|               car|                bike|   30
9xx|        GA|  1.0|        0.0|      01-04-1999|           5.0|     3.0|    0.0|   1687.0|       9.4|      4.0|
f|      0.0|          0.0|    1008.71|        1008.71|         456.46|       435.17|               0.0|    117.08|
1.11|  01-04-2013|         119.66|        01-09-2013|                        0.0|         0.0|    80935.0|         2380
0.0|          1|
|1076863|  1277178|    10000|      10000|        10000.0| 36 months|   13.49|     339.31|    C|       C1| AIR RESOURCES
BOARD| 10+ years|          RENT|   49200.0|    Source Verified|01-12-2011|             other|            personel|   91
7xx|        CA| 20.0|        0.0|      01-02-1996|           1.0|    10.0|    0.0|   5598.0|      21.0|     37.0|
f|      0.0|          0.0|12226.30221|        12226.3|        10000.0|      2209.33|             16.97|       0.0|
0.0|  01-01-2015|         357.48|        01-01-2015|                        0.0|         0.0|    80935.0|         23800.
0|          0|
|1075358|  1311748|     3000|       3000|         3000.0| 60 months|   12.69|      67.79|    B|       B5|University Med
ica...|    1 year|          RENT|   80000.0|    Source Verified|01-12-2011|             other|            Personal|   9
72xx|        OR|17.94|        0.0|      01-01-1996|           0.0|    15.0|    0.0|  27783.0|      53.9|     38.0|
f|    766.9|        766.9|    3242.17|        3242.17|         2233.1|      1009.07|               0.0|       0.0|
0.0|  01-01-2016|          67.79|        01-01-2016|                        0.0|         0.0|    80935.0|         23800.
0|          0|
|1075269|  1311441|     5000|       5000|         5000.0| 36 months|    7.9|     156.46|    A|       A4|Veolia Transpo
rtaton|   3 years|          RENT|   36000.0|    Source Verified|01-12-2011|           wedding|My wedding loan I...|   8
52xx|        AZ| 11.2|        0.0|      01-11-2004|           3.0|     9.0|    0.0|   7963.0|      28.3|     12.0|
f|      0.0|          0.0|5631.377753|        5631.38|         5000.0|       631.38|               0.0|       0.0|
0.0|  01-01-2015|         161.03|        01-09-2015|                        0.0|         0.0|    80935.0|         23800.
0|          0|
|1069639|  1304742|     7000|       7000|         7000.0| 60 months|   15.96|     170.08|    C|       C5|Southern Star
Pho...|   8 years|          RENT|   47004.0|       Not Verified|01-12-2011|debt_consolidation|                Loan|   2
80xx|        NC|23.51|        0.0|      01-07-2005|           1.0|     7.0|    0.0|  17726.0|      85.6|     11.0|
f|  1889.15|      1889.15|    8136.84|        8136.84|         5110.85|      3025.99|               0.0|       0.0|
0.0|  01-01-2016|         170.08|        01-01-2016|                        0.0|         0.0|    80935.0|         23800.
0|          0|
+-------+---------+---------+----------+---------------+----------+--------+----------+-----+---------+--------------
------+----------+-------------+----------+-------------------+----------+------------------+--------------------+----
----+----------+-----+----------+---------------+-------------+--------+-------+---------+----------+---------+-----
-------------+---------+-------------+----------+---------------+---------------+-------------+------------------+---
-------+----------------------+-----------+---------------+-----------------+-------------------------+-----------
+-----------+---------------+-----------+
only showing top 5 rows
```

In [ ]:

```
df = df.withColumn("home_ownership",
                   when(col("home_ownership") == "NONE", "OTHER")
                   .when(col("home_ownership") == "ANY", "OTHER")
                   .otherwise(col("home_ownership")))
```

Build Transformation Pipeline

In [ ]:

```
categorical_columns = ['term', 'grade', 'home_ownership', 'verification_status', 'purpose', 'sub_grade', 'addr_state']

# Define a UDF to extract a specific index from a SparseVector
def extract_from_vector(vec, i):
    try:
        return int(vec.toArray()[i])
    except IndexError:
        return 0

# Register the UDF
extract_from_vector_udf = udf(extract_from_vector, IntegerType())

# Filter categorical columns with fewer than 7 distinct categories
filtered_columns = [c for c in categorical_columns if df.select(c).distinct().count() < 7]

# Prepare stages for pipeline
indexers = [StringIndexer(inputCol=c, outputCol=f"{c}_Index", handleInvalid="keep") for c in filtered_columns]
encoders = [OneHotEncoder(inputCols=[f"{c}_Index"], outputCols=[f"{c}_OHE"], dropLast=False) for c in filtered_columns]

# Build the pipeline
pipeline = Pipeline(stages=indexers + encoders)
model = pipeline.fit(df)
transformed_df = model.transform(df)

# Convert one-hot encoded vectors to boolean columns for each category
for i, indexer in enumerate(indexers):
    column = indexer.getInputCol().replace('_Index', '')
    categories = model.stages[i].labels  # Fetch category labels from the indexer
    for j, category in enumerate(categories):
        # Create a new column for each category, using the UDF to extract the value from the vector
        transformed_df = transformed_df.withColumn(f"{column}_{category.replace(' ', '_')}", extract_from_vector_udf(co
    transformed_df = transformed_df.drop(f"{column}_OHE", f"{column}_Index")  # Clean up the DataFrame

# Show the transformed DataFrame structure
transformed_df.show(5, truncate=False)
```

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

```
+-----------------+----------+--------------------+-----------+--------------+-----------------+-----------------+--------
--------+------------+----------+----------------+----------+--------------+---------------+----------------------
+-------------------+-------------------+--------------------+-----------------------------------+----------------------
---------+-------------------------+
|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   |purpose           |title
|zip_code|addr_state|dti  |delinq_2yrs|earliest_cr_line|inq_last_6mths|open_acc|pub_rec|revol_bal|revol_util|total_acc|
initial_list_status|out_prncp|out_prncp_inv|total_pymnt|total_pymnt_inv|total_rec_prncp|total_rec_int|total_rec_late_fe
e|recoveries|collection_recovery_fee|last_pymnt_d|last_pymnt_amnt|last_credit_pull_d|collections_12_mths_ex_med|tot_col
l_amt|tot_cur_bal|total_rev_hi_lim|default_ind|term__36_months|term__60_months|home_ownership_MORTGAGE|home_ownership_R
ENT|home_ownership_OWN|home_ownership_OTHER|verification_status_Source_Verified|verification_status_Not_Verified|verifi
cation_status_Verified|
+-------+--------+---------+-----------+---------------+---------+--------+-----------+-----+---------+--------------
-----------+----------+-------------+----------+-----------------+----------+------------------+-------------------
-----------------+--------+----------+-----+-----------+-----------------+----------+-------------+--------+-------+---------+----
------+---------+------------------+---------+-------------+-----------+---------------+-----------------+-------------+
-----------------+-----------+-----------+-----------------+-----------+---------------+---------------+----------------------
--------+------------+----------+----------------+----------+--------------+---------------+----------------------
+-------------------+-------------------+--------------------+-----------------------------------+----------------------
---------+-------------------------+
|1077430|1314167  |2500     |2500       |2500.0         | 60 months|15.27   |59.83      |C    |C4       |Ryder
|< 1 year  |RENT          |30000.0   |Source Verified    |01-12-2011|car               |bike
|309xx   |GA        |1.0  |0.0        |01-04-1999      |5.0           |3.0     |0.0    |1687.0   |9.4       |4.0      |
f                  |0.0      |0.0          |1008.71    |1008.71        |456.46         |435.17       |0.0
|117.08    |1.11                    |01-04-2013  |119.66         |01-09-2013        |0.0                        |0.0
|80935.0    |23800.0         |1          |0             |1              |0                      |1                  |0
|0                  |1                                  |0                                  |0
|
|1076863|1277178  |10000    |10000      |10000.0        | 36 months|13.49   |339.31     |C    |C1       |AIR RESOURCES
BOARD      |10+ years |RENT          |49200.0   |Source Verified    |01-12-2011|other             |personel
|917xx   |CA        |20.0 |0.0        |01-02-1996      |1.0           |10.0    |0.0    |5598.0   |21.0      |37.0     |
f                  |0.0      |0.0          |12226.30221|12226.3        |10000.0        |2209.33      |16.97
|0.0       |0.0                     |01-01-2015  |357.48         |01-01-2015        |0.0                        |0.0
|80935.0    |23800.0         |0          |1              |0              |0                      |1                  |0
|0                  |1                                  |0                                  |0
|
|1075358|1311748  |3000     |3000       |3000.0         | 60 months|12.69   |67.79      |B    |B5       |University Med
ical Group |1 year    |RENT          |80000.0   |Source Verified    |01-12-2011|other             |Personal
|972xx   |OR        |17.94|0.0        |01-01-1996      |0.0           |15.0    |0.0    |27783.0  |53.9      |38.0     |
f                  |766.9    |766.9        |3242.17    |3242.17        |2233.1         |1009.07      |0.0
|0.0       |0.0                     |01-01-2016  |67.79          |01-01-2016        |0.0                        |0.0
|80935.0    |23800.0         |0          |0             |1              |0                      |1                  |0
|0                  |1                                  |0                                  |0
|
|1075269|1311441  |5000     |5000       |5000.0         | 36 months|7.9     |156.46     |A    |A4       |Veolia Transpo
rtaton     |3 years   |RENT          |36000.0   |Source Verified    |01-12-2011|wedding           |My wedding loan I pr
omise to pay back|852xx   |AZ        |11.2 |0.0        |01-11-2004      |3.0           |9.0     |0.0    |7963.0   |28.3
```

```
|12.0     |f                 |0.0      |0.0          |5631.377753|5631.38        |5000.0         |631.38       |0.0
|0.0      |0.0                    |01-01-2015  |161.03         |01-09-2015        |0.0                          |0.0
|80935.0    |23800.0         |0          |1               |0              |0                          |1                  |0
|0                   |1                                  |0                                  |0
|
|1069639|1304742  |7000     |7000       |7000.0         | 60 months|15.96   |170.08     |C    |C5       |Southern Star
Photography|8 years   |RENT          |47004.0   |Not Verified       |01-12-2011|debt_consolidation|Loan
|280xx   |NC        |23.51|0.0        |01-07-2005      |1.0           |7.0     |0.0    |17726.0  |85.6      |11.0     |
f                  |1889.15  |1889.15      |8136.84    |8136.84        |5110.85        |3025.99      |0.0
|0.0       |0.0                    |01-01-2016  |170.08         |01-01-2016        |0.0                          |0.0
|80935.0    |23800.0         |0          |0               |1              |0                          |1                  |0
|0                   |0                                  |1                                  |0
|
+-------+--------+--------+----------+--------------+---------+-------+----------+-----+--------+-------------
----------+---------+-------------+---------+------------------+----------+---------+-----------------+--------------------
----------------+--------+---------+-----+----------+--------------+-------------+-------+-------+---------+----
------+---------+------------------+--------+-----------+----------+--------------+--------------+-------------+
-----------------+----------+---------------------+-----------+---------------+-----------------+-----------------
--------+-----------+----------+---------------+----------+-------------+-------------+---------------------
+------------------+-----------------+-------------------+--------------------------------+--------------------
---------+--------------------------+
only showing top 5 rows
```

In [ ]:

```
# Drop the specified columns
transformed_df = transformed_df.drop("home_ownership", "verification_status", "term")

# Show the structure of the modified DataFrame
transformed_df.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- member_id: integer (nullable = true)
 |-- loan_amnt: integer (nullable = true)
 |-- funded_amnt: integer (nullable = true)
 |-- funded_amnt_inv: double (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)
 |-- annual_inc: double (nullable = true)
 |-- issue_d: 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)
|-- inq_last_6mths: 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)
|-- last_credit_pull_d: string (nullable = true)
|-- collections_12_mths_ex_med: double (nullable = true)
|-- tot_coll_amt: double (nullable = true)
|-- tot_cur_bal: double (nullable = true)
|-- total_rev_hi_lim: double (nullable = true)
|-- default_ind: integer (nullable = true)
|-- term__36_months: integer (nullable = true)
|-- term__60_months: integer (nullable = true)
|-- home_ownership_MORTGAGE: integer (nullable = true)
|-- home_ownership_RENT: integer (nullable = true)
|-- home_ownership_OWN: integer (nullable = true)
|-- home_ownership_OTHER: integer (nullable = true)
|-- verification_status_Source_Verified: integer (nullable = true)
|-- verification_status_Not_Verified: integer (nullable = true)
|-- verification_status_Verified: integer (nullable = true)
```

In [ ]:

```
# List of all columns except 'default_ind'
columns = [col for col in df.columns if col != 'default_ind']

# Append 'default_ind' to the end of the list
columns.append('default_ind')
```

```
# Rearrange the DataFrame
df = df.select(*columns)
```

Feature Selection by calculating Correlation Matrix for target variable

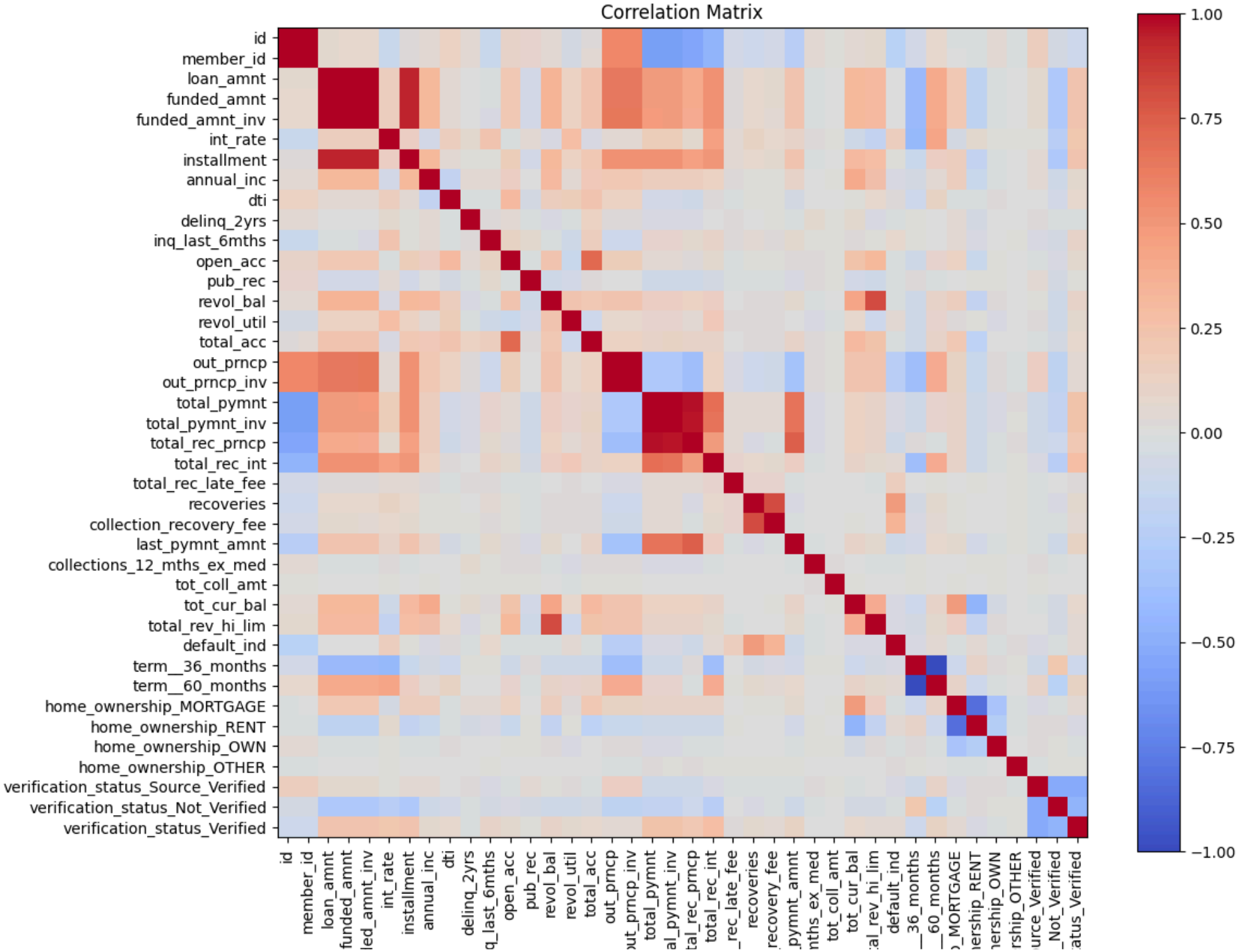
In [ ]:

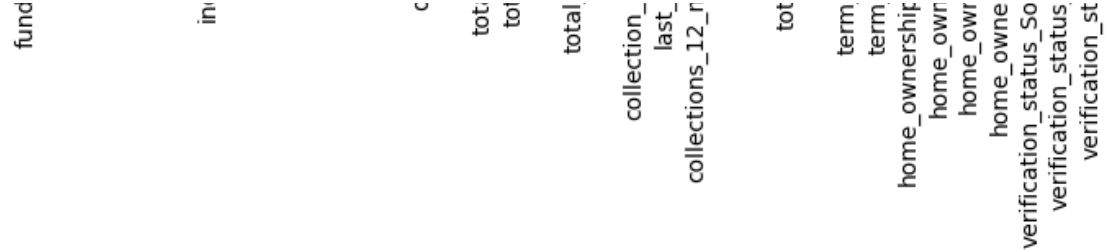
```
# Identify numeric columns only
numeric_cols = [col for col in transformed_df.columns if isinstance(transformed_df.schema[col].dataType, (IntegerType,

# Assemble the numeric features into a single vector column
assembler = VectorAssembler(inputCols=numeric_cols, outputCol="features")
transformed_vector = assembler.transform(transformed_df)

# Get correlation matrix
correlation_matrix = Correlation.corr(transformed_vector, "features", "pearson").head()

# Extract the correlation values
correlation_values = correlation_matrix[0].toArray()

# Find indices of the target feature for correlation, assuming target feature is in the numeric_cols
target_index = numeric_cols.index('default_ind')

# Get correlations with the target
target_correlations = correlation_values[target_index]

# Create list of (feature, correlation) tuples
feature_correlations = list(zip(numeric_cols, target_correlations))

# Sort features by absolute correlation with the target
sorted_features = sorted(feature_correlations, key=lambda x: abs(x[1]), reverse=True)

# Most and Least relevant features
most_relevant_features = sorted_features[:15]
least_relevant_features = sorted_features[-15:]

print("Most Relevant Features:")
for feature, corr in most_relevant_features:
    print(f"{feature}: {corr}")

print("\nLeast Relevant Features:")
for feature, corr in least_relevant_features:
    print(f"{feature}: {corr}")
```

```
Most Relevant Features:
default_ind: 1.0
recoveries: 0.48080265355718005
collection_recovery_fee: 0.33699004310396785
out_prncp: -0.22347217766058045
out_prncp_inv: -0.223471081994506
member_id: -0.21741507367504356
id: -0.2170605894354415
int_rate: 0.15563491442792285
total_rec_late_fee: 0.14231926960158683
total_rec_prncp: -0.09055219851601892
last_pymnt_amnt: -0.08754057732481188
inq_last_6mths: 0.0733862377428352
verification_status_Verified: 0.05158720369869314
total_rec_int: 0.048793434525129964
tot_cur_bal: -0.04545843510264604

Least Relevant Features:
revol_bal: -0.020900747805252386
pub_rec: -0.019739708239587248
total_acc: -0.01905579368206222
open_acc: -0.019024766274326487
verification_status_Not_Verified: -0.015765583916530462
dti: 0.012953023277306042
collections_12_mths_ex_med: -0.010779294773893965
delinq_2yrs: -0.00954584944071254
home_ownership_OTHER: 0.008721262667963138
home_ownership_OWN: -0.008067799392603233
installment: 0.006462360344170498
funded_amnt_inv: -0.0059627935198225375
funded_amnt: -0.003670578718554105
loan_amnt: -0.0027911273559767936
tot_coll_amt: -0.0023703716776752246
```

In [ ]:

```
# Plotting the correlation matrix with labels
plt.figure(figsize=(12, 10))
plt.imshow(correlation_values, interpolation='nearest', cmap='coolwarm')
plt.colorbar()
plt.title('Correlation Matrix')
plt.xticks(ticks=np.arange(len(numeric_cols)), labels=numeric_cols, rotation=90)
plt.yticks(ticks=np.arange(len(numeric_cols)), labels=numeric_cols)
plt.show()
```


Correlation Matrix
id
member_id
loan_amnt
funded_amnt
funded_amnt_inv
int_rate
installment
annual_inc
dti
delinq_2yrs
inq_last_6mths
open_acc
pub_rec
revol_bal
revol_util
total_acc
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_amnt
collections_12_mths_ex_med
tot_coll_amt
tot_cur_bal
total_rev_hi_lim
default_ind
term__36_months
term__60_months
home_ownership_MORTGAGE
home_ownership_RENT
home_ownership_OWN
home_ownership_OTHER
verification_status_Source_Verified
verification_status_Not_Verified
verification_status_Verified
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



Feature Selection using RandomForestClassifier

In [ ]:

```
# Identify numeric columns only
numeric_cols = [col for col in transformed_df.columns if isinstance(transformed_df.schema[col].dataType, (IntegerType,

# Assemble features into a feature vector
assembler = VectorAssembler(
    inputCols=numeric_cols,
    outputCol="features"
)

# Initialize the Random Forest model
rf = RandomForestClassifier(featuresCol="features", labelCol="default_ind")

# Create a pipeline
pipeline = Pipeline(stages=[assembler, rf])

# Fit the model
model = pipeline.fit(transformed_df)

# Get feature importances
importances = model.stages[-1].featureImportances

# Zip feature names with their importances
feature_importance_list = list(zip(numeric_cols, importances))

# Sort features by importance
sorted_features = sorted(feature_importance_list, key=lambda x: x[1], reverse=True)

# Output the sorted feature list
print("Features sorted by importance:")
for feature, importance in sorted_features:
    print(f"{feature}: {importance}")
```

```
Features sorted by importance:
recoveries: 0.3130381301614628
collection_recovery_fee: 0.2229735510786283
total_rec_prncp: 0.09842829371395921
last_pymnt_amnt: 0.05844796642477594
out_prncp_inv: 0.05782354719054393
total_pymnt: 0.044765418680427604
funded_amnt: 0.037351985289899574
out_prncp: 0.03616609589688329
installment: 0.025545307363609642
id: 0.02410369508094905
funded_amnt_inv: 0.022502505087929447
total_pymnt_inv: 0.015591436214003735
member_id: 0.01464344945621285
total_rec_late_fee: 0.007871073200638052
loan_amnt: 0.007391355774744599
int_rate: 0.005762514293555615
term__36_months: 0.0037169340978588303
term__60_months: 0.002066830134391874
total_rec_int: 0.0006789094635909348
annual_inc: 0.00046819179465948726
verification_status_Verified: 0.0002850336982954546
tot_cur_bal: 0.0002393741238996366
revol_bal: 4.852103823731774e-05
revol_util: 4.560227602573505e-05
dti: 3.2508545213591906e-05
total_rev_hi_lim: 9.902181436132885e-06
total_acc: 1.6588582229453605e-06
inq_last_6mths: 2.0887994466637753e-07
delinq_2yrs: 0.0
open_acc: 0.0
pub_rec: 0.0
collections_12_mths_ex_med: 0.0
tot_coll_amt: 0.0
home_ownership_MORTGAGE: 0.0
home_ownership_RENT: 0.0
home_ownership_OWN: 0.0
home_ownership_OTHER: 0.0
verification_status_Source_Verified: 0.0
verification_status_Not_Verified: 0.0
```

In [ ]:
```
transformed_df = transformed_df.drop("id", "total_acc", "collections_12_mths_ex_med", "title", "open_acc", "pub_rec", "
```

In [ ]:
```
transformed_df.show(5)
```

```
+---------+-----------+---------------+--------+-----------+-----+---------+----------+----------+----------+----------
----+--------+----------+-----+-----------+----------------+--------------+---------+----------+-------------------+---
------+-------------+-----------+---------------+---------------+-------------+------------------+----------+----------
-------------+------------+---------------+------------------+-----------+----------------+---------------+------------
---+-----------------------+-------------------+------------------+--------------------+-------------------------------
----+----------------------------+--------------------------------+-----------+
|loan_amnt|funded_amnt|funded_amnt_inv|int_rate|installment|grade|sub_grade|emp_length|annual_inc|   issue_d|       pur
pose|zip_code|addr_state|  dti|delinq_2yrs|earliest_cr_line|inq_last_6mths|revol_bal|revol_util|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|last_credit_pull_d|tot_cur_bal|total_rev_hi_lim|term__36_months|term__60_mon
ths|home_ownership_MORTGAGE|home_ownership_RENT|home_ownership_OWN|home_ownership_OTHER|verification_status_Source_Veri
fied|verification_status_Verified|verification_status_Not_Verified|default_ind|
+---------+-----------+---------------+--------+-----------+-----+---------+----------+----------+----------+----------
----+--------+----------+-----+-----------+----------------+--------------+---------+----------+-------------------+---
------+-------------+-----------+---------------+---------------+-------------+------------------+----------+----------
-------------+------------+---------------+------------------+-----------+----------------+---------------+------------
---+-----------------------+-------------------+------------------+--------------------+-------------------------------
----+----------------------------+--------------------------------+-----------+
|     5000|       5000|         4975.0|   10.65|     162.87|    B|       B2| 10+ years|   24000.0|01-12-2011|   credit_
card|   860xx|        AZ|27.65|        0.0|      01-01-1985|           1.0|  13648.0|      83.7|                  f|
0.0|          0.0|5861.071414|        5831.78|         5000.0|       861.07|                 0.0|       0.0|
0.0|  01-01-2015|         171.62|        01-01-2016|    80935.0|         23800.0|              1|              0|
0|                  1|                 0|                   0|                                  0|
1|                              0|          0|
|     2500|       2500|         2500.0|   15.27|      59.83|    C|       C4|  < 1 year|   30000.0|01-12-2011|
car|   309xx|        GA|  1.0|        0.0|      01-04-1999|           5.0|   1687.0|       9.4|                  f|
0.0|          0.0|    1008.71|        1008.71|         456.46|       435.17|                 0.0|    117.08|
1.11|  01-04-2013|         119.66|        01-09-2013|    80935.0|         23800.0|              0|              1|
0|                  1|                 0|                   0|                                  1|
0|                              0|          1|
|     2400|       2400|         2400.0|   15.96|      84.33|    C|       C5| 10+ years|   12252.0|01-12-2011|small_busi
ness|   606xx|        IL| 8.72|        0.0|      01-11-2001|           2.0|   2956.0|      98.5|                  f|
0.0|          0.0|3003.653644|        3003.65|         2400.0|       603.65|                 0.0|       0.0|
0.0|  01-06-2014|         649.91|        01-01-2016|    80935.0|         23800.0|              1|              0|
0|                  1|                 0|                   0|                                  0|
0|                              1|          0|
|    10000|      10000|        10000.0|   13.49|     339.31|    C|       C1| 10+ years|   49200.0|01-12-2011|         o
ther|   917xx|        CA| 20.0|        0.0|      01-02-1996|           1.0|   5598.0|      21.0|                  f|
0.0|          0.0|12226.30221|        12226.3|        10000.0|      2209.33|               16.97|       0.0|
0.0|  01-01-2015|         357.48|        01-01-2015|    80935.0|         23800.0|              1|              0|
0|                  1|                 0|                   0|                                  1|
0|                              0|          0|
|     3000|       3000|         3000.0|   12.69|      67.79|    B|       B5|    1 year|   80000.0|01-12-2011|         o
ther|   972xx|        OR|17.94|        0.0|      01-01-1996|           0.0|  27783.0|      53.9|                  f|
766.9|        766.9|    3242.17|        3242.17|         2233.1|      1009.07|                 0.0|       0.0|
0.0|  01-01-2016|          67.79|        01-01-2016|    80935.0|         23800.0|              0|              1|
0|                  1|                 0|                   0|                                  1|
```

```
0|                                0|          0|
+---------+-----------+--------------+--------+-----------+-----+---------+---------+----------+----------+----------
----+--------+----------+-----+-----------+-----------------+--------------+---------+----------+-------------------+---
------+-------------+-----------+---------------+---------------+-------------+-----------------+----------+----------
-------------+-----------+---------------+------------------+----------+----------------+---------------+-----------
---+---------------------+-------------------+-----------------+--------------------+----------------------------
----+--------------------------+-----------------------------------+-----------+
only showing top 5 rows
```

Feature Scaling

In [ ]:

```
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.sql.functions import udf, col
from pyspark.ml.linalg import DenseVector
from pyspark.sql.types import ArrayType, DoubleType

# List of numerical columns for scaling, adjusted based on the existing columns
numerical_columns = ['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'int_rate', 'installment', 'annual_inc',
                     'dti', 'delinq_2yrs', 'inq_last_6mths', 'revol_bal', 'revol_util', '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_amnt',
                     'tot_cur_bal', 'total_rev_hi_lim']

# Assemble numerical features into a single vector
assembler = VectorAssembler(inputCols=numerical_columns, outputCol="unscaled_features")
transformed_df = assembler.transform(df)

# Apply StandardScaler to the assembled features
scaler = StandardScaler(inputCol="unscaled_features", outputCol="scaled_features", withMean=True, withStd=True)
scaler_model = scaler.fit(transformed_df)
transformed_df = scaler_model.transform(transformed_df)

# Define a UDF to convert Vector to array of doubles
def vector_to_array(v):
    return v.toArray().tolist()

vector_to_array_udf = udf(vector_to_array, ArrayType(DoubleType()))

# Convert the scaled_features column from vector to array
transformed_df = transformed_df.withColumn("scaled_features_array", vector_to_array_udf(col("scaled_features")))

# Overwrite the original columns with the scaled values
```

```
for i, col_name in enumerate(numerical_columns):
    transformed_df = transformed_df.withColumn(col_name, col("scaled_features_array")[i])

# Drop the intermediate columns
transformed_df = transformed_df.drop("unscaled_features", "scaled_features", "scaled_features_array")
```

In [ ]:

```
transformed_df.show(5)
```

```
+-------------------+-------------------+-------------------+-------------------+-------------------+-----+---------+
---------+-------------------+----------+--------------+--------+----------+--------------------+------------------+-
---------------+-----------------+--------------------+--------------------+--------------------+--------------------+--
---------------+-------------------+--------------------+--------------------+--------------------+-----------------
---+-------------------+-------------------------+-----------+--------------------+--------------------+-----------------
--+-------------------+-------------+-------------+---------------------+------------------+-----------------+-
------------------+-----------------------------------+-------------------------------+-------------------------------+-
----------+
|          loan_amnt|        funded_amnt|    funded_amnt_inv|           int_rate|         installment|grade|sub_grade|
emp_length|         annual_inc|   issue_d|       purpose|zip_code|addr_state|                dti|        delinq_2yrs|e
arliest_cr_line|    inq_last_6mths|           revol_bal|          revol_util|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|last_credit_pull_d|        tot_cur_bal|   total_re
v_hi_lim|term__36_months|term__60_months|home_ownership_MORTGAGE|home_ownership_RENT|home_ownership_OWN|home_ownership_
OTHER|verification_status_Source_Verified|verification_status_Verified|verification_status_Not_Verified|default_ind|
+-------------------+-------------------+-------------------+-------------------+-------------------+-----+---------+
---------+-------------------+----------+--------------+--------+----------+--------------------+------------------+-
---------------+-----------------+--------------------+--------------------+--------------------+--------------------+--
---------------+-------------------+--------------------+--------------------+--------------------+-----------------
---+-------------------+-------------------------+-----------+--------------------+--------------------+-----------------
--+-------------------+-------------+-------------+---------------------+------------------+-----------------+-
------------------+-----------------------------------+-------------------------------+-------------------------------+-
----------+
|-1.1574978635011264|-1.1567338604948179|-1.1551262372144446| -0.5832902470210084| -1.1225194486407128|    B|       B2|
10+ years|-0.7974836225584876|01-12-2011|   credit_card|   860xx|        AZ|  0.5464063537174678|-0.3633185873049472|
01-01-1985|0.3303204231049117|-0.14693349804902422|  1.2035001104001952|                  f|-0.9757652974672304|-0.9757
631019862665|-0.23642840523610226|-0.23697520462148122|-0.13629838979538744| -0.4376191880216884|-0.08885306081555977|-
0.11435811806698513|   -0.07966576706600416|  01-01-2015| -0.4253110659636703|        01-01-2016|-0.3646833583594943|-
0.2118903002675386|              1|              0|                    0|                 1|                0|
0|                                  0|                                  1|                                  0|         0|
| -1.454400957194763|-1.4538450995182943|-1.4490577945797103|  0.4741505600851613| -1.5454662719361265|    C|       C4|
< 1 year|-0.7037258060188002|01-12-2011|           car|   309xx|        GA| -0.9801821897978946|-0.3633185873049472|
01-04-1999|4.4775401821971705| -0.6850975177242207| -1.9182189315096259|                  f|-0.9757652974672304|-0.9757
631019862665| -0.8497370313811486| -0.8484401950929014| -0.8160675350326854| -0.6418994465429331|-0.08885306081555977|
0.1714042573093129|   -0.06171589795806208|  01-04-2013|-0.43594771287603384|        01-09-2013|-0.3646833583594943|-0.
2118903002675386|              0|              1|                    0|                 1|                0|
0|                                  1|                                  0|                                  0|         1|
|-1.4662770809425083|-1.4657295490792335| -1.460933817099519|  0.6320800312763428| -1.4449014702014833|    C|       C5|
```

```
10+ years|-0.9810614273431956|01-12-2011|small_business|   606xx|        IL| -0.5379584165919434|-0.3633185873049472|
01-11-2001|1.3671253628779765| -0.6280011093162428|   1.825323015114399|                  f|-0.9757652974672304|-0.9757
631019862665| -0.5975884296697526| -0.5955232836462387| -0.5252901842149313| -0.5610890733199717|-0.08885306081555977|-
0.11435811806698513|   -0.07966576706600416|  01-06-2014| -0.3274010996104303|        01-01-2016|-0.3646833583594943|-
0.2118903002675386|              1|              0|                     0|                  1|                 0|
0|                                 0|                          0|                                  1|         0|
|-0.5636916761138532|-0.5625113824478648|-0.5583561055940571| 0.06673830107023018|-0.39828868871987855|    C|       C1|
10+ years|-0.4037007930918003|01-12-2011|         other|   917xx|        CA| 0.10819238156390232|-0.3633185873049472|
01-02-1996|0.3303204231049117| -0.5091289966181516| -1.4308442224093039|                  f|-0.9757652974672304|-0.9757
631019862665|  0.5680976021069484|  0.5737169708744799|  0.6117627533191201|  0.2090652481226022|   4.598426851510363|-
0.11435811806698513|   -0.07966576706600416|  01-01-2015| -0.3872639682898465|        01-01-2015|-0.3646833583594943|-
0.2118903002675386|              1|              0|                     0|                  1|                 0|
0|                                 1|                          0|                                  0|         0|
|-1.3950203384560356| -1.394422851713599|-1.3896776819806667|-0.11636833219490762|   -1.51279297308683|    B|       B5|
1 year|0.07758933181192856|01-12-2011|         other|   972xx|        OR|-0.00981033512058...|-0.3633185873049472|
01-01-1996|-0.706484516668153|  0.4890458028546546|-0.04854871125408042|                  f|-0.8848079509333091|-0.8847
689551902732| -0.5674414298627257| -0.5652839071646113| -0.5502604651720936|-0.36663191809648477|-0.08885306081555977|-
0.11435811806698513|   -0.07966576706600416|  01-01-2016|-0.44656593603508493|        01-01-2016|-0.3646833583594943|-
0.2118903002675386|              0|              1|                     0|                  1|                 0|
0|                                 1|                          0|                                  0|         0|
+------------------+-------------------+-------------------+--------------------+--------------------+-----+---------+
---------+-------------------+----------+--------------+--------+----------+--------------------+--------------------+-
--------------+-------------------+--------------------+--------------------+--------------------+--------------------+--
----------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------
---+--------------------+----------------------+------------+--------------------+--------------------+----------------
--+--------------------+--------------+--------------+----------------------+--------------------+--------------------+-
------------------+-----------------------------------+--------------------------+-------------------------------+-
----------+
only showing top 5 rows
```

Split dataset into 80% train, 10% val, 10% test

In [ ]:

```
# Randomly split the DataFrame into training, validation, and testing sets
train_df, val_df, test_df = transformed_df.randomSplit([0.8, 0.1, 0.1], seed=42)

# Show the row count for each DataFrame to verify the split
print("Training Set Row Count:", train_df.count())
print("Validation Set Row Count:", val_df.count())
print("Testing Set Row Count:", test_df.count())
```

```
Training Set Row Count: 676909
Validation Set Row Count: 84420
Testing Set Row Count: 84988
```

In [ ]:
```
# Save the training set
train_df.coalesce(1).write.csv(path="/content/drive/MyDrive/train_data", mode='overwrite', header=True)

# Save the validation set
val_df.coalesce(1).write.csv(path="/content/drive/MyDrive/validation_data", mode='overwrite', header=True)

# Save the testing set
test_df.coalesce(1).write.csv(path="/content/drive/MyDrive/test_data", mode='overwrite', header=True)
```

In [ ]:
```
transformed_df.coalesce(1).write.csv(path="/content/drive/MyDrive/dataV2.csv", mode='overwrite', header=True)
```

Stage2-Modelling

In [ ]:
```
#Create Spark session
spark = SparkSession.builder.getOrCreate()
spark.sparkContext.setLogLevel("Error")
```

Data Loading

In [ ]:
```
# Read Spark dataframe
train_data = spark.read.csv("/content/drive/MyDrive/UOW/316/316-project/data/train_data.csv", inferSchema=True, header=
test_data = spark.read.csv("/content/drive/MyDrive/UOW/316/316-project/data/validation_data.csv", inferSchema=True, hea
val_data = spark.read.csv("/content/drive/MyDrive/UOW/316/316-project/data/test_data.csv", inferSchema=True, header=Tru

# Convert Spark DataFrames to Pandas DataFrames
train_data_pd = train_data.toPandas()
val_data_pd = val_data.toPandas()
test_data_pd = test_data.toPandas()
```

Dataset format for Neural Network

In [ ]:
```
def encode_columns(df):
    for col_info in df.dtypes:
        if col_info[1] == 'string':
            indexer = StringIndexer(inputCol=col_info[0], outputCol=f"{col_info[0]}_encoded")
```

```
            df = indexer.fit(df).transform(df)
            df = df.drop(col_info[0])
    return df
```

In [ ]:

```
# Splitting each DataFrame into features and labels
# For training data
X_train = encode_columns(train_data).drop("member_id").toPandas()
y_train = train_data_pd[['default_ind']] # Select only the target column for labels

# For validation data
X_val = encode_columns(val_data).drop("member_id").toPandas()
y_val = val_data_pd[['default_ind']]

# For testing data
X_test = encode_columns(test_data).drop("member_id").toPandas()
y_test = test_data_pd[['default_ind']]

print("Shape of X_train:", X_train.shape)
print("Shape of y_train:", y_train.shape)

print("Shape of X_val:", X_val.shape)
print("Shape of y_val:", y_val.shape)

print("Shape of X_test:", X_test.shape)
print("Shape of y_test:", y_test.shape)
```

Dataset format for GB and RF

In [ ]:

```
# Convert pandas dataframe to TensorFlow dataframe (For GB and RF)
tfdf_train = tfdf.keras.pd_dataframe_to_tf_dataset(train_data_pd, label="default_ind")
tfdf_val= tfdf.keras.pd_dataframe_to_tf_dataset(val_data_pd, label="default_ind")
tfdf_test = tfdf.keras.pd_dataframe_to_tf_dataset(test_data_pd, label="default_ind")
```

Tensorflow

Training

Gradient Boosted Tree

In [ ]:

```
## Gradient Boosted Tree
gbt = tfdf.keras.GradientBoostedTreesModel(
    task=tfdf.keras.Task.CLASSIFICATION,
    num_trees=5,
    max_depth=10
)
gbt.compile(metrics=["accuracy"])

# Train the model
history=gbt.fit(tfdf_train, validation_data=tfdf_val)

# Evaluate the model on the val set
val_results = gbt.evaluate(tfdf_val)

# Print the accuracy
accuracy = val_results[1]
print("Accuracy:", accuracy)
```

```
Use /tmp/tmpuvjsvtbj as temporary training directory
Reading training dataset...
WARNING:tensorflow:6 out of the last 6 calls to <function CoreModel._consumes_training_examples_until_eof at 0x7c05a2e8
ea70> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) cr
eating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead
of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=Tr
ue option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#cont
rolling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
Training dataset read in 0:00:18.906212. Found 676909 examples.
Reading validation dataset...
Num validation examples: tf.Tensor(84988, shape=(), dtype=int32)
Validation dataset read in 0:00:02.413227. Found 84988 examples.
Training model...
Model trained in 0:01:29.210878
Compiling model...
WARNING:tensorflow:5 out of the last 5 calls to <function InferenceCoreModel.make_predict_function.<locals>.predict_fun
ction_trained at 0x7c056bc07d00> triggered tf.function retracing. Tracing is expensive and the excessive number of trac
ings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) pas
sing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.func
tion has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensor
flow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more deta
ils.
Model compiled.
85/85 [==============================] - 2s 24ms/step - loss: 0.0000e+00 - accuracy: 0.9962
Accuracy: 0.9961641430854797
```

Random Forest Tree

In [ ]:
```
###################### Random Forest Tree ######################
rft_model = tfdf.keras.RandomForestModel(
    task=tfdf.keras.Task.CLASSIFICATION,
    num_trees=5,
    max_depth=10)

rft_model.compile(metrics=["accuracy"])

# Train the model
rft_model.fit(tfdf_train)
```

```
Use /tmp/tmpp87gm2pb as temporary training directory
Reading training dataset...
Training dataset read in 0:00:17.914852. Found 676909 examples.
Training model...
Model trained in 0:19:27.641824
Compiling model...
Model compiled.
```

Out[ ]:
```
<tf_keras.src.callbacks.History at 0x789b84d844f0>
```

Neural Network

In [ ]:
```
##################### Neural Network #####################

# define the keras model
def build_model(n_hidden=1, n_neurons=30, input_shape=(X_train.shape[1],)):

    model = Sequential()
    model.add(Dense(n_neurons,activation='relu', input_shape=input_shape))

    # Vary the number of hidden layers and number of neurons in each layer
    for _ in range(n_hidden-1):
        model.add(Dense(n_neurons, activation="relu"))

    # Fix the output layer
    model.add(Dense(1, activation="sigmoid"))

    # Fix the loss function and metrics
    model.compile(optimizer='Adam',
              loss='binary_crossentropy',
```

```
                metrics=['accuracy'])
    return model

# build the keras model
nn_model = build_model(n_hidden=4, n_neurons=64, input_shape=(X_train.shape[1],))
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)
# Fit the Keras model
nn_model.fit(x=X_train,
          y=y_train,
          epochs=40,
          verbose=1,
          validation_data=(X_val, y_val),
          callbacks=[early_stop])
```

```
Epoch 1/40
21154/21154 [==============================] - 47s 2ms/step - loss: 2.3149 - accuracy: 0.9718 - val_loss: 0.1971 - val_
accuracy: 0.9508
Epoch 2/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.1604 - accuracy: 0.9664 - val_loss: 0.0645 - val_
accuracy: 0.9895
Epoch 3/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.1198 - accuracy: 0.9774 - val_loss: 0.1471 - val_
accuracy: 0.9662
Epoch 4/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.1280 - accuracy: 0.9776 - val_loss: 0.1124 - val_
accuracy: 0.9760
Epoch 5/40
21154/21154 [==============================] - 43s 2ms/step - loss: 0.0900 - accuracy: 0.9844 - val_loss: 0.0944 - val_
accuracy: 0.9808
Epoch 6/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.1018 - accuracy: 0.9791 - val_loss: 0.0488 - val_
accuracy: 0.9918
Epoch 7/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0752 - accuracy: 0.9900 - val_loss: 0.0378 - val_
accuracy: 0.9946
Epoch 8/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0639 - accuracy: 0.9940 - val_loss: 0.0462 - val_
accuracy: 0.9920
Epoch 9/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0488 - accuracy: 0.9925 - val_loss: 0.0516 - val_
accuracy: 0.9908
Epoch 10/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0369 - accuracy: 0.9949 - val_loss: 0.0220 - val_
accuracy: 0.9981
Epoch 11/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0286 - accuracy: 0.9964 - val_loss: 0.0130 - val_
accuracy: 0.9982
```

```
Epoch 12/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0269 - accuracy: 0.9961 - val_loss: 0.0248 - val_
accuracy: 0.9962
Epoch 13/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0333 - accuracy: 0.9950 - val_loss: 0.0155 - val_
accuracy: 0.9976
Epoch 14/40
21154/21154 [==============================] - 47s 2ms/step - loss: 0.0325 - accuracy: 0.9960 - val_loss: 0.0792 - val_
accuracy: 0.9846
Epoch 15/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.0400 - accuracy: 0.9940 - val_loss: 0.0281 - val_
accuracy: 0.9956
Epoch 16/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.0277 - accuracy: 0.9962 - val_loss: 0.0156 - val_
accuracy: 0.9977
Epoch 17/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.0324 - accuracy: 0.9966 - val_loss: 0.0209 - val_
accuracy: 0.9976
Epoch 18/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.0303 - accuracy: 0.9966 - val_loss: 0.0121 - val_
accuracy: 0.9984
Epoch 19/40
21154/21154 [==============================] - 47s 2ms/step - loss: 0.0220 - accuracy: 0.9968 - val_loss: 0.0121 - val_
accuracy: 0.9984
Epoch 20/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0211 - accuracy: 0.9974 - val_loss: 0.0129 - val_
accuracy: 0.9982
Epoch 21/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0228 - accuracy: 0.9968 - val_loss: 0.0229 - val_
accuracy: 0.9965
Epoch 22/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0214 - accuracy: 0.9970 - val_loss: 0.0187 - val_
accuracy: 0.9972
Epoch 23/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0208 - accuracy: 0.9970 - val_loss: 0.0151 - val_
accuracy: 0.9979
Epoch 24/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0189 - accuracy: 0.9974 - val_loss: 0.0187 - val_
accuracy: 0.9982
Epoch 25/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0271 - accuracy: 0.9976 - val_loss: 0.0110 - val_
accuracy: 0.9985
Epoch 26/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0432 - accuracy: 0.9976 - val_loss: 0.0667 - val_
accuracy: 0.9946
Epoch 27/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0834 - accuracy: 0.9748 - val_loss: 0.0439 - val_
```

```
accuracy: 0.9952
Epoch 28/40
21154/21154 [==============================] - 46s 2ms/step - loss: 0.0387 - accuracy: 0.9941 - val_loss: 0.0195 - val_
accuracy: 0.9968
Epoch 29/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0329 - accuracy: 0.9921 - val_loss: 0.0388 - val_
accuracy: 0.9936
Epoch 30/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0328 - accuracy: 0.9928 - val_loss: 0.0226 - val_
accuracy: 0.9974
Epoch 31/40
21154/21154 [==============================] - 43s 2ms/step - loss: 0.0403 - accuracy: 0.9915 - val_loss: 0.1016 - val_
accuracy: 0.9456
Epoch 32/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0490 - accuracy: 0.9865 - val_loss: 0.0244 - val_
accuracy: 0.9967
Epoch 33/40
21154/21154 [==============================] - 43s 2ms/step - loss: 0.0299 - accuracy: 0.9953 - val_loss: 0.0142 - val_
accuracy: 0.9982
Epoch 34/40
21154/21154 [==============================] - 44s 2ms/step - loss: 0.0272 - accuracy: 0.9956 - val_loss: 0.0158 - val_
accuracy: 0.9979
Epoch 35/40
21154/21154 [==============================] - 45s 2ms/step - loss: 0.0290 - accuracy: 0.9952 - val_loss: 0.0153 - val_
accuracy: 0.9980
Epoch 35: early stopping
```

Out[ ]:
```
<keras.callbacks.History at 0x7bbffd1d2c50>
```

Evaluation

Gradient Boosted Tree

In [ ]:
```
################## Gradient Boosted Tree ##################
gbt.make_inspector().evaluation()

################ Evaluate on Validation Data ###############
probs = (gbt.predict(tfdf_val))
y_pred = probs.round(0)

# Extract the true labels from the validation dataset
y_true = np.concatenate([y for _, y in tfdf_val.as_numpy_iterator()])

# Calculate precision, recall, and ROC
precision = precision_score(y_true, y_pred)
recall = recall_score(y_true, y_pred)
```

```
roc_auc = roc_auc_score(y_true, probs)

# Print precision, recall, and ROC
print("Precision:", precision)
print("Recall:", recall)
print("ROC AUC:", roc_auc)
print()


############### Evaluate on Test Data ###############
# Make predictions on the test set
predictions = gbt.predict(tfdf_test)

# Print out the first few predictions to understand the structure
print(predictions[:5])

#Convert probabilities to class labels (using 0.5 as threshold)
y_pred = (predictions > 0.5).astype(int).flatten()

# Convert true labels to numpy array
y_true = test_data_pd["default_ind"].values

# Confusion matrix
conf_matrix = confusion_matrix(y_true, y_pred)
print("Confusion Matrix:")
print(conf_matrix)
print()

# Classification report
class_report = classification_report(y_true, y_pred)
print("Classification Report:")
print(class_report)
print()
```

```
85/85 [==============================] - 2s 26ms/step
Precision: 0.9991256830601093
Recall: 0.988538062283737
ROC AUC: 0.9996580855011945

85/85 [==============================] - 4s 47ms/step
[[2.5153703e-03]
 [6.1177392e-04]
 [2.1951289e-01]
 [9.0897046e-03]
 [9.9490535e-01]]
```

```
Confusion Matrix:
[[79842     1]
 [   69  4508]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     79843
           1       1.00      0.98      0.99      4577

    accuracy                           1.00     84420
   macro avg       1.00      0.99      1.00     84420
weighted avg       1.00      1.00      1.00     84420
```

Random Forest Tree

In [ ]:

```
################## Random Forest Tree ##################

################ Evaluate on Validation Data ###############
# Evaluate the model on validation data
val_predictions = rft_model.predict(tfdf_val)

# Extract the true labels and predicted probabilities
true_labels_val = val_data_pd["default_ind"]
predicted_probabilities_val = val_predictions[:, 0]  # Assuming the model returns probabilities for the positive class

# Convert probabilities to binary predictions with a threshold of 0.5
predicted_labels_val = (predicted_probabilities_val >= 0.5).astype(int)

# Calculate evaluation metrics for validation data
val_accuracy = accuracy_score(true_labels_val, predicted_labels_val)
val_precision = precision_score(true_labels_val, predicted_labels_val, zero_division=0)
val_recall = recall_score(true_labels_val, predicted_labels_val, zero_division=0)
val_roc_auc = roc_auc_score(true_labels_val, predicted_probabilities_val)

# Print evaluation metrics for validation data
print(f"Validation Data Accuracy: {val_accuracy}")
print(f"Validation Data Precision: {val_precision}")
print(f"Validation Data Recall: {val_recall}")
print(f"Validation Data ROC AUC: {val_roc_auc}")

# Print the confusion matrix for validation data
val_conf_matrix = confusion_matrix(true_labels_val, predicted_labels_val)
```

```
print(f"Validation Data Confusion Matrix:\n{val_conf_matrix}")



############### Evaluate on Test Data ###############
# Evaluate the model on test data
evaluation = rft_model.evaluate(tfdf_test)
print(f"Test Data Evaluation: {evaluation}")

# Make predictions on test data
test_predictions = rft_model.predict(tfdf_test)

# Extract the true labels and predicted probabilities
true_labels_test = test_data_pd["default_ind"]
predicted_probabilities_test = test_predictions[:, 0]  # Assuming the model returns probabilities for the positive clas

# Convert probabilities to binary predictions with a threshold of 0.5
predicted_labels_test = (predicted_probabilities_test >= 0.5).astype(int)

# Calculate evaluation metrics for test data
test_accuracy = accuracy_score(true_labels_test, predicted_labels_test)
test_precision = precision_score(true_labels_test, predicted_labels_test, zero_division=0)
test_recall = recall_score(true_labels_test, predicted_labels_test, zero_division=0)
test_roc_auc = roc_auc_score(true_labels_test, predicted_probabilities_test)

# Print evaluation metrics for test data
print(f"Test Data Accuracy: {test_accuracy}")
print(f"Test Data Precision: {test_precision}")
print(f"Test Data Recall: {test_recall}")
print(f"Test Data ROC AUC: {test_roc_auc}")

# Print the confusion matrix for test data
test_conf_matrix = confusion_matrix(true_labels_test, predicted_labels_test)
print(f"Test Data Confusion Matrix:\n{test_conf_matrix}")
```

```
85/85 [==============================] - 5s 57ms/step
Validation Data Accuracy: 0.9992822516119922
Validation Data Precision: 1.0
Validation Data Recall: 0.9868079584775087
Validation Data ROC AUC: 0.9998784630816463
Validation Data Confusion Matrix:
[[80364     0]
 [   61  4563]]
85/85 [==============================] - 6s 67ms/step - loss: 0.0000e+00
Test Data Evaluation: 0.0
```

```
85/85 [==============================] - 5s 56ms/step
Test Data Accuracy: 0.9989931295901445
Test Data Precision: 1.0
Test Data Recall: 0.9814288835481757
Test Data ROC AUC: 0.9999876710743107
Test Data Confusion Matrix:
[[79843     0]
 [   85  4492]]
```

Neural Network

In [ ]:

```
test_results = nn_model.evaluate(X_test, y_test, verbose=0)

print("Test Loss:", test_results[0])
print("Test Accuracy:", test_results[1])
```

```
Test Loss: 0.028894230723381042
Test Accuracy: 0.9934493899345398
```

Tuning

In [ ]:

```
!pip install tensorflow==2.12.0
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
```

In [ ]:

```
import tensorflow_decision_forests as tfdf

class ModelTuner:
    def __init__(self, model_type):
        self.model_type = model_type
        self.model = None
        self.grid_search = None

    def tune_parameters(self, X_train, y_train, param_grid):
        if self.model_type == 'GBT':

            # Add discretized hyperparameters for the tuner
            tuner = tfdf.tuner.RandomSearch(num_trials=3)
            tuner.choice("max_depth", param_grid.get("max_depth", []))
            tuner.choice("num_trees", param_grid.get("num_trees", []))

            # Create GBT model with a random search tuner
            self.model = tfdf.keras.GradientBoostedTreesModel(
```

```
            task=tfdf.keras.Task.CLASSIFICATION,
            tuner=tuner
        )
        self.model.compile(metrics=["accuracy"])

        # Cannot use val when tuning according to documentation
        self.model.fit(X_train)

        # Display the tuning logs.
        tuning_logs = self.model.make_inspector().tuning_logs()

        print()
        print("Best Hyperparameters")
        print(tuning_logs[tuning_logs.best].iloc[0])

        # Evaluate the model on the val set
        val_results = self.model.evaluate(y_train)

        # Print the accuracy
        accuracy = val_results[1]
        print("Accuracy:", accuracy)

        return tuning_logs

    elif self.model_type == 'NN':
        # Create and tune a Neural Network model using KerasClassifier wrapped with GridSearchCV
        self.model = KerasClassifier(build_fn=build_model, verbose=0)
        self.grid_search = GridSearchCV(self.model, param_grid, cv=3, scoring='accuracy')
        self.grid_search.fit(X_train, y_train, batch_size=32, validation_data=(X_val, y_val))
        print("Best Parameters:", self.grid_search.best_params_)
        return self.grid_search.best_params_

    elif self.model_type == 'RF':
        # Add discretized hyperparameters for the tuner
        tuner = tfdf.tuner.RandomSearch(num_trials=3)
        tuner.choice("max_depth", param_grid.get("max_depth", []))
        tuner.choice("num_trees", param_grid.get("num_trees", []))

        # Create RF model with a random search tuner
        self.model = tfdf.keras.RandomForestModel(
            task=tfdf.keras.Task.CLASSIFICATION,
            tuner=tuner
        )
```

```
            self.model.compile(metrics=["accuracy"])

             # Cannot use val when tuning according to documentation
            self.model.fit(X_train)

            # Display the tuning logs.
            tuning_logs = self.model.make_inspector().tuning_logs()

            print()
            print("Best Hyperparameters")
            print(tuning_logs[tuning_logs.best].iloc[0])

            # Evaluate the model on the val set
            val_results = self.model.evaluate(y_train)

            # Print the accuracy
            accuracy = val_results[1]
            print("Accuracy:", accuracy)

            return tuning_logs

        else:
            raise ValueError("Unsupported model type")
```

Gradient Boosted Tree

In [ ]:

```
model_tuner = ModelTuner(model_type='GBT')
param_grid = {
    "num_trees": [10, 15],
    "max_depth": [5, 10]
}
tunning_logs = model_tuner.tune_parameters(X_train=tfdf_train, y_train=tfdf_val, param_grid=param_grid)
```

```
Use /tmp/tmpnxzfkiuu as temporary training directory
Reading training dataset...
Training dataset read in 0:00:24.304041. Found 676909 examples.
Training model...
Model trained in 0:09:05.484934
Compiling model...
Model compiled.
WARNING:tensorflow:6 out of the last 10 calls to <function InferenceCoreModel.yggdrasil_model_path_tensor at 0x7c056a44
ea70> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) cr
eating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead
of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=Tr
```

```
ue option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#cont
rolling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
Best Hyperparameters
score               -0.051649
evaluation_time    544.429521
best                     True
max_depth                  10
num_trees                  15
Name: 2, dtype: object
85/85 [==============================] - 3s 26ms/step - loss: 0.0000e+00 - accuracy: 0.9984
Accuracy: 0.9983644485473633
```

Random Forest Tree

In [ ]:

```
model_tuner = ModelTuner(model_type='RF')
param_grid = {
    "num_trees": [10, 15],
    "max_depth": [5, 10]
}
rf_tunning_logs = model_tuner.tune_parameters(X_train=tfdf_train, y_train=tfdf_val, param_grid=param_grid)
```

```
Use /tmp/tmpsuemwvrm as temporary training directory
Reading training dataset...
Training dataset read in 0:00:48.127731. Found 676909 examples.
Training model...
Model trained in 0:02:59.810163
Compiling model...
Model compiled.

Best Hyperparameters
score                0.996474
evaluation_time    177.317566
best                     True
max_depth                  10
num_trees                  15
Name: 2, dtype: object
85/85 [==============================] - 11s 35ms/step - loss: 0.0000e+00 - accuracy: 0.9978
Accuracy: 0.9977761507034302
```

Neural Network

In [ ]:

```
model_tuner = ModelTuner(model_type='NN')
param_grid = {
    'n_hidden': [4, 8],
    'n_neurons': [64, 128]
```

```
}
best_params = model_tuner.tune_parameters(X_train, y_train, param_grid)
best_model = build_model(**best_params)
history = best_model.fit(X_train, y_train, epochs=300, batch_size=32, verbose=0, validation_data=(X_test, y_test),
          callbacks=[early_stop])
```

```
10584/10584 [==============================] - 40s 4ms/step - loss: 62.1395 - accuracy: 0.9933 - val_loss: 7.3726 - val
_accuracy: 0.9953
10585/10585 [==============================] - 24s 2ms/step - loss: 12.0875 - accuracy: 0.9925
10585/10585 [==============================] - 46s 4ms/step - loss: 38.7610 - accuracy: 0.9858 - val_loss: 13.0068 - va
l_accuracy: 0.9960
10584/10584 [==============================] - 23s 2ms/step - loss: 7.2344 - accuracy: 0.9988
10584/10584 [==============================] - 33s 3ms/step - loss: 21.0015 - accuracy: 0.9954 - val_loss: 14.0542 - va
l_accuracy: 0.9960
10585/10585 [==============================] - 24s 2ms/step - loss: 20.4832 - accuracy: 0.9937
10585/10585 [==============================] - 40s 4ms/step - loss: 21.7450 - accuracy: 0.9871 - val_loss: 4.7520 - val
_accuracy: 0.9974
10584/10584 [==============================] - 23s 2ms/step - loss: 3.9096 - accuracy: 0.9985
10584/10584 [==============================] - 35s 3ms/step - loss: 24.4246 - accuracy: 0.9952 - val_loss: 23.4436 - va
l_accuracy: 0.9957
10585/10585 [==============================] - 24s 2ms/step - loss: 21.5479 - accuracy: 0.9933
10585/10585 [==============================] - 39s 4ms/step - loss: 34.0695 - accuracy: 0.9863 - val_loss: 4.5447 - val
_accuracy: 0.9840
10584/10584 [==============================] - 22s 2ms/step - loss: 5.8461 - accuracy: 0.9738
10584/10584 [==============================] - 43s 4ms/step - loss: 3.5357 - accuracy: 0.9942 - val_loss: 0.1488 - val_
accuracy: 0.9779
10585/10585 [==============================] - 27s 3ms/step - loss: 0.2154 - accuracy: 0.9613
10585/10585 [==============================] - 55s 5ms/step - loss: 4.9580 - accuracy: 0.9635 - val_loss: 0.1840 - val_
accuracy: 0.9692
10584/10584 [==============================] - 24s 2ms/step - loss: 0.1908 - accuracy: 0.9936
10584/10584 [==============================] - 45s 4ms/step - loss: 5.4119 - accuracy: 0.9944 - val_loss: 0.1467 - val_
accuracy: 0.9904
10585/10585 [==============================] - 26s 2ms/step - loss: 0.2408 - accuracy: 0.9828
10585/10585 [==============================] - 50s 5ms/step - loss: 4.9602 - accuracy: 0.9634 - val_loss: 0.2253 - val_
accuracy: 0.9459
10584/10584 [==============================] - 25s 2ms/step - loss: 0.2099 - accuracy: 0.9867
10584/10584 [==============================] - 50s 5ms/step - loss: 4.6533 - accuracy: 0.9943 - val_loss: 0.0960 - val_
accuracy: 0.9930
10585/10585 [==============================] - 27s 2ms/step - loss: 0.1461 - accuracy: 0.9882
10585/10585 [==============================] - 51s 5ms/step - loss: 4.3104 - accuracy: 0.9818 - val_loss: 0.0718 - val_
accuracy: 0.9951
10584/10584 [==============================] - 25s 2ms/step - loss: 0.0460 - accuracy: 0.9982
10584/10584 [==============================] - 56s 5ms/step - loss: 0.2911 - accuracy: 0.9923 - val_loss: 0.0524 - val_
accuracy: 0.9834
10585/10585 [==============================] - 29s 3ms/step - loss: 0.0827 - accuracy: 0.9703
10585/10585 [==============================] - 64s 6ms/step - loss: 0.4886 - accuracy: 0.9814 - val_loss: 0.0237 - val_
accuracy: 0.9969
```

```
10584/10584 [==============================] - 26s 2ms/step - loss: 0.0166 - accuracy: 0.9987
10584/10584 [==============================] - 64s 6ms/step - loss: 0.5767 - accuracy: 0.9934 - val_loss: 0.1097 - val_
accuracy: 0.9816
10585/10585 [==============================] - 28s 3ms/step - loss: 0.1830 - accuracy: 0.9674
10585/10585 [==============================] - 63s 6ms/step - loss: 0.7360 - accuracy: 0.9811 - val_loss: 0.0314 - val_
accuracy: 0.9962
10584/10584 [==============================] - 27s 3ms/step - loss: 0.0192 - accuracy: 0.9984
10584/10584 [==============================] - 59s 5ms/step - loss: 1.3176 - accuracy: 0.9914 - val_loss: 0.0522 - val_
accuracy: 0.9944
10585/10585 [==============================] - 29s 3ms/step - loss: 0.0784 - accuracy: 0.9901
10585/10585 [==============================] - 69s 6ms/step - loss: 0.7957 - accuracy: 0.9816 - val_loss: 0.0265 - val_
accuracy: 0.9948
10584/10584 [==============================] - 27s 3ms/step - loss: 0.0153 - accuracy: 0.9987
21169/21169 [==============================] - 77s 4ms/step - loss: 18.6434 - accuracy: 0.9914 - val_loss: 3.3928 - val
_accuracy: 0.9975
```

In [ ]:

```
test_results = best_model.evaluate(X_test, y_test, verbose=0)

print("Test Loss:", test_results[0])
print("Test Accuracy:", test_results[1])
```

```
Test Loss: 0.034795477986335754
Test Accuracy: 0.9978758692741394
```

In [ ]:

```
# Get the best parameter
print("Best parameters found: ", best_params)
```

```
Best parameters found:  {'n_hidden': 1, 'n_neurons': 128}
```

PySpark

In [ ]:

```
from pyspark.ml.classification import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from pyspark.ml import Pipeline
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
import matplotlib.pyplot as plt
import seaborn as sns

import pyspark
from pyspark.sql import SparkSession
from pyspark.ml.linalg import DenseVector
```

```
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import StringIndexer
from pyspark.ml.classification import MultilayerPerceptronClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator
```

Preprocessing

In [ ]:

```
# Select features
# Define the feature columns (excluding 'member_id' and 'default_ind')
feature_cols = [
 'loan_amnt',
 'funded_amnt',
 'funded_amnt_inv',
 'int_rate',
 'installment',
 'grade',
 'sub_grade',
 'emp_length',
 'annual_inc',
 'issue_d',
 'purpose',
 'zip_code',
 'addr_state',
 'dti',
 'delinq_2yrs',
 'earliest_cr_line',
 'inq_last_6mths',
 'revol_bal',
 'revol_util',
 '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',
 'last_credit_pull_d',
```

```
 'tot_cur_bal',
 'total_rev_hi_lim',
 'term__36_months',
 'term__60_months',
 'home_ownership_MORTGAGE',
 'home_ownership_RENT',
 'home_ownership_OWN',
 'home_ownership_OTHER',
 'verification_status_Source_Verified',
 'verification_status_Verified',
 'verification_status_Not_Verified'
]

# Separate numeric and string columns
numeric_cols = [
 'loan_amnt',
 'funded_amnt',
 'funded_amnt_inv',
 'int_rate',
 'installment',
 'annual_inc',
 'dti',
 'delinq_2yrs',
 'inq_last_6mths',
 'revol_bal',
 'revol_util',
 '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_amnt',
 'tot_cur_bal',
 'total_rev_hi_lim',
 'term__36_months',
 'term__60_months',
 'home_ownership_MORTGAGE',
 'home_ownership_RENT',
 'home_ownership_OWN',
 'home_ownership_OTHER',
```

```
 'verification_status_Source_Verified',
 'verification_status_Verified',
 'verification_status_Not_Verified'
]

string_cols = [
 'grade',
 'sub_grade',
 'emp_length',
 'issue_d',
 'purpose',
 'zip_code',
 'addr_state',
 'earliest_cr_line',
 'initial_list_status',
 'last_pymnt_d',
 'last_credit_pull_d'
]

# Handle numeric columns and string columns
# Index and encode string columns
indexers = [StringIndexer(inputCol=col, outputCol=col + "_index", handleInvalid="keep") for col in string_cols]
encoders = [OneHotEncoder(inputCol=col + "_index", outputCol=col + "_encoded") for col in string_cols]
```

In [ ]:

```
# Assemble all features
assembler = VectorAssembler(inputCols=numeric_cols + [col + "_encoded" for col in string_cols], outputCol="features")

# Index the target variable
indexer = StringIndexer(inputCol="default_ind", outputCol="label", handleInvalid="keep")
```

Training and Tuning

Neural Network

In [ ]:

```
# Define the layers for the MLP model
input_size = len(train_data.head().features)
layers = [input_size, 64, 32, 2]

# Instantiate the MLP model
mlp_classifier = MultilayerPerceptronClassifier(labelCol='label',
                                                featuresCol='features',
                                                maxIter=100,
```

```
                                                    layers=layers,
                                                    seed=1234)
```

In [ ]:

```
# Define the parameter grid for hyperparameter tuning
paramGrid = (ParamGridBuilder()
             .addGrid(mlp_classifier.layers, [[input_size, 64, 2], [input_size, 128, 2], [input_size, 256, 2]])
             .addGrid(mlp_classifier.solver, ['gd', 'l-bfgs'])
             .build())

# Define the CrossValidator
crossval = CrossValidator(estimator=mlp_classifier,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", met
                          numFolds=3)
```

In [ ]:

```
# Early stopping criteria
early_stopping_rounds = 3
best_score = 0.0
patience = 0

# Custom cross-validation loop
param_maps = crossval.getEstimatorParamMaps()
nn_spark_model = None
for i, param_map in enumerate(param_maps):
    # Set the parameters for the current iteration
    model = mlp_classifier.copy(param_map)
    # Perform cross-validation
    cv_model = model.fit(train_data)

    # Evaluate the model
    predictions = cv_model.transform(val_data)
    evaluator = MulticlassClassificationEvaluator(labelCol="label",
                                                  predictionCol="prediction",
                                                  metricName="accuracy")
    accuracy = evaluator.evaluate(predictions)

    print(f'Param map: {param_maps}')
    print(f"Iteration {i+1}/{len(param_maps)}, Accuracy: {accuracy}")

    # Check for early stopping
    if accuracy > best_score:
```

```
        best_score = accuracy
        nn_spark_model = cv_model
        patience = 0
    else:
        patience += 1

    if patience >= early_stopping_rounds:
        print("Early stopping triggered")
        break
```

```
Iteration 1/9, Accuracy: 0.9907012556266288
Iteration 2/9, Accuracy: 0.9907012556266288
Iteration 3/9, Accuracy: 0.9907012556266288
Iteration 4/9, Accuracy: 0.9990168206586117
Iteration 5/9, Accuracy: 0.9990642027955461
Iteration 6/9, Accuracy: 0.9990405117270789
Iteration 7/9, Accuracy: 0.9989694385216773
Iteration 8/9, Accuracy: 0.999360341151386
Iteration 9/9, Accuracy: 0.9992892679459844
```

Decision Tree

In [ ]:

```
# Define the Decision Tree Classifier
dt = DecisionTreeClassifier(featuresCol="features", labelCol="label")

# Create a pipeline to assemble the steps
pipeline = Pipeline(stages=indexers + encoders + [assembler, indexer, dt])

# Define the parameter grid for hyperparameter tuning
paramGrid = (ParamGridBuilder()
             .addGrid(dt.maxDepth, [5, 10, 15])
             .addGrid(dt.minInstancesPerNode, [1, 5, 10])
             .build())

# Define the CrossValidator
crossval = CrossValidator(estimator=pipeline,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", met
                          numFolds=3)
```

In [ ]:

```
# Early stopping criteria
early_stopping_rounds = 3
best_score = 0.0
```

```
patience = 0

# Custom cross-validation loop
param_maps = crossval.getEstimatorParamMaps()
dt_spark_model = None

for i, param_map in enumerate(param_maps):
    # Set the parameters for the current iteration
    model = pipeline.copy(param_map)
    # Perform cross-validation
    cv_model = model.fit(train_data)

    # Evaluate the model
    predictions = cv_model.transform(val_data)
    evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
    accuracy = evaluator.evaluate(predictions)

    print(f"Iteration {i+1}/{len(param_maps)}, Accuracy: {accuracy}")

    # Check for early stopping
    if accuracy > best_score:
        best_score = accuracy
        dt_spark_model = cv_model
        patience = 0
    else:
        patience += 1

    if patience >= early_stopping_rounds:
      print("Early stopping triggered")
      break
```

```
Iteration 1/9, Accuracy: 0.9907012556266288
Iteration 2/9, Accuracy: 0.9907012556266288
Iteration 3/9, Accuracy: 0.9907012556266288
Iteration 4/9, Accuracy: 0.9990168206586117
Iteration 5/9, Accuracy: 0.9990642027955461
Iteration 6/9, Accuracy: 0.9990405117270789
Iteration 7/9, Accuracy: 0.9989694385216773
Iteration 8/9, Accuracy: 0.999360341151386
Iteration 9/9, Accuracy: 0.9992892679459844
```

Random Forest

In [ ]:

```
# Import Random Forest Classifier
from pyspark.ml.classification import RandomForestClassifier
```

```
# Initialize the Random Forest model
rf = RandomForestClassifier(featuresCol="features", labelCol='label')

# Define a parameter grid to search through
param_grid = (ParamGridBuilder()
              .addGrid(rf.maxDepth, [5, 10, 15])
              .addGrid(rf.numTrees, [10, 20, 30])
              .build())

# Define evaluator
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")

# Define cross-validator
cross_val = CrossValidator(estimator=rf,
                           estimatorParamMaps=param_grid,
                           evaluator=evaluator,
                           numFolds=3)

# Fit cross-validator to train data
cv_model = cross_val.fit(train_data)

# Get best model from cross-validation
best_rf_model = cv_model.bestModel

# Print model parameters
print("Model Parameters:")
for param, value in best_rf_model.extractParamMap().items():
    print(f"{param}: {value}")
```

```
Model Parameters:
RandomForestClassifier_2587b6c40a10__bootstrap: True
RandomForestClassifier_2587b6c40a10__cacheNodeIds: False
RandomForestClassifier_2587b6c40a10__checkpointInterval: 10
RandomForestClassifier_2587b6c40a10__featureSubsetStrategy: auto
RandomForestClassifier_2587b6c40a10__featuresCol: features
RandomForestClassifier_2587b6c40a10__impurity: gini
RandomForestClassifier_2587b6c40a10__labelCol: default_ind
RandomForestClassifier_2587b6c40a10__leafCol:
RandomForestClassifier_2587b6c40a10__maxBins: 32
RandomForestClassifier_2587b6c40a10__maxDepth: 5
RandomForestClassifier_2587b6c40a10__maxMemoryInMB: 256
RandomForestClassifier_2587b6c40a10__minInfoGain: 0.0
RandomForestClassifier_2587b6c40a10__minInstancesPerNode: 1
RandomForestClassifier_2587b6c40a10__minWeightFractionPerNode: 0.0
```

```
RandomForestClassifier_2587b6c40a10__numTrees: 20
RandomForestClassifier_2587b6c40a10__predictionCol: prediction
RandomForestClassifier_2587b6c40a10__probabilityCol: probability
RandomForestClassifier_2587b6c40a10__rawPredictionCol: rawPrediction
RandomForestClassifier_2587b6c40a10__seed: 8765275973112923211
RandomForestClassifier_2587b6c40a10__subsamplingRate: 1.0
```

In [ ]:

Evaluation

Neural Network

In [ ]:
```
# Predict the model on test set
predictions = nn_spark_model.transform(test_data)
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Overall accuracy rate:", accuracy)
```

```
Overall accuracy rate: 0.9919900320398718
```

Decision Tree

In [ ]:
```
# Make predictions on the validation data
predictions = dt_spark_model.transform(val_data)

# Evaluate the model
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)

print(f"Best model accuracy: {accuracy}")
```

```
Best model accuracy: 0.9992892679459844
```

Random Forest

In [ ]:
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator, MulticlassClassificationEvaluator

# Make predictions on the test and validation data using the best model
test_predictions_tuned = best_rf_model.transform(test_data)
val_predictions_tuned = best_rf_model.transform(val_data)
```

```
# Define the threshold for binary classification
threshold = 0.5
target_column = "default_ind"

# Define evaluators
multiclass_evaluator = MulticlassClassificationEvaluator(labelCol=target_column)

# Define a function to calculate precision and recall manually
def calculate_precision_recall(predictions, threshold):
    true_positives = predictions.filter((predictions[target_column] == 1) & (predictions["prediction"] == 1)).count()
    false_positives = predictions.filter((predictions[target_column] == 0) & (predictions["prediction"] == 1)).count()
    false_negatives = predictions.filter((predictions[target_column] == 1) & (predictions["prediction"] == 0)).count()

    precision = true_positives / (true_positives + false_positives)
    recall = true_positives / (true_positives + false_negatives)

    return precision, recall

# Evaluate the tuned model on the test set
test_auc_tuned = multiclass_evaluator.evaluate(test_predictions_tuned, {multiclass_evaluator.metricName: "areaUnderROC"
test_accuracy_tuned = multiclass_evaluator.evaluate(test_predictions_tuned, {multiclass_evaluator.metricName: "accuracy
test_precision_tuned, test_recall_tuned = calculate_precision_recall(test_predictions_tuned, threshold)
test_f1_tuned = 2 * (test_precision_tuned * test_recall_tuned) / (test_precision_tuned + test_recall_tuned)

print("Tuned Model Evaluation on Test Set:")
print(f"AUC: {test_auc_tuned}")
print(f"Accuracy: {test_accuracy_tuned}")
print(f"Precision: {test_precision_tuned}")
print(f"Recall: {test_recall_tuned}")
print(f"F1 Score: {test_f1_tuned}")
```

```
Tuned Model Evaluation on Test Set:
AUC: 0.9976899804742231
Accuracy: 0.9979526521391255
Precision: 0.9995509654243376
Recall: 0.96280276816609
F1 Score: 0.9808327825512226
```

In [ ]:

```
# Stop the Spark session
spark.stop()
```