Lending Club Loan Data Analysis

Course-end Project 2

DESCRIPTION

Create a model that predicts whether or not a loan will be default using the historical data.

Problem Statement:

For companies like Lending Club correctly predicting whether or not a loan will be a default is very important. In this project, using the historical data from 2007 to 2015, you have to build a deep learning model to predict the chance of default for future loans. As you will see later this dataset is highly imbalanced and includes a lot of features that make this problem more challenging.

Domain: Finance

Analysis to be done: Perform data preprocessing and build a deep learning prediction model.

Content:

Dataset columns and definition:

credit.policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.

purpose: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").

int.rate: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be more risky are assigned higher interest rates.

installment: The monthly installments owed by the borrower if the loan is funded.

log.annual.inc: The natural log of the self-reported annual income of the borrower.

dti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).

fico: The FICO credit score of the borrower.

days.with.cr.line: The number of days the borrower has had a credit line.

revol.bal: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).

revol.util: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).

inq.last.6mths: The borrower's number of inquiries by creditors in the last 6 months.

delinq.2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.

pub.rec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).

Steps to perform:

Perform exploratory data analysis and feature engineering and then apply feature engineering. Follow up with a deep learning model to predict whether or not the loan will be default using the historical data.

Tasks:

1. `Feature Transformation`

Transform categorical values into numerical values (discrete)

1. `Exploratory data analysis of different factors of the dataset.`
2. `Additional Feature Engineering`

You will check the correlation between features and will drop those features which have a strong correlation

This will help reduce the number of features and will leave you with the most relevant features

1. `Modeling`

After applying EDA and feature engineering, you are now ready to build the predictive models

In this part, you will create a deep learning model using Keras with Tensorflow backend

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Reading data

```
data = pd.read_csv("/content/drive/MyDrive/Elective 1 - Deep
Learning/Projects/Project 3 - Lending Club Loan Data
Analysis/loan_data.csv")
data.head()
```

```
   credit.policy             purpose  int.rate  installment
log.annual.inc  \
0              1  debt_consolidation    0.1189       829.10
11.350407
1              1         credit_card    0.1071       228.22
11.082143
2              1  debt_consolidation    0.1357       366.86
10.373491
3              1  debt_consolidation    0.1008       162.34
11.350407
4              1         credit_card    0.1426       102.92
11.299732

     dti  fico  days.with.cr.line  revol.bal  revol.util
inq.last.6mths  \
0  19.48   737        5639.958333      28854        52.1
0
1  14.29   707        2760.000000      33623        76.7
0
2  11.63   682        4710.000000       3511        25.6
1
3   8.10   712        2699.958333      33667        73.2
1
4  14.97   667        4066.000000       4740        39.5
0

   delinq.2yrs  pub.rec  not.fully.paid
0            0        0               0
1            0        0               0
2            0        0               0
3            0        0               0
4            1        0               0
```

Checking for null values

```
data.isnull().sum()
```

```
credit.policy        0
purpose              0
int.rate             0
installment          0
log.annual.inc       0
dti                  0
fico                 0
days.with.cr.line    0
revol.bal            0
```

```
revol.util           0
inq.last.6mths       0
delinq.2yrs          0
pub.rec              0
not.fully.paid       0
dtype: int64
```

```
data.isna().sum()
```

```
credit.policy        0
purpose              0
int.rate             0
installment          0
log.annual.inc       0
dti                  0
fico                 0
days.with.cr.line    0
revol.bal            0
revol.util           0
inq.last.6mths       0
delinq.2yrs          0
pub.rec              0
not.fully.paid       0
dtype: int64
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   credit.policy      9578 non-null   int64  
 1   purpose            9578 non-null   object 
 2   int.rate           9578 non-null   float64
 3   installment        9578 non-null   float64
 4   log.annual.inc     9578 non-null   float64
 5   dti                9578 non-null   float64
 6   fico               9578 non-null   int64  
 7   days.with.cr.line  9578 non-null   float64
 8   revol.bal          9578 non-null   int64  
 9   revol.util         9578 non-null   float64
 10  inq.last.6mths     9578 non-null   int64  
 11  delinq.2yrs        9578 non-null   int64  
 12  pub.rec            9578 non-null   int64  
 13  not.fully.paid     9578 non-null   int64  
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```

```
data["purpose"].value_counts()
```

```
debt_consolidation    3957
all_other             2331
credit_card           1262
home_improvement       629
small_business         619
major_purchase         437
educational            343
Name: purpose, dtype: int64
```

Transform categorical values into numerical values

```
data = pd.get_dummies(data, columns = ["purpose"])
data.head()
```

```
   credit.policy  int.rate  installment  log.annual.inc    dti
fico  \
0              1    0.1189       829.10       11.350407  19.48   737

1              1    0.1071       228.22       11.082143  14.29   707

2              1    0.1357       366.86       10.373491  11.63   682

3              1    0.1008       162.34       11.350407   8.10   712

4              1    0.1426       102.92       11.299732  14.97   667


   days.with.cr.line  revol.bal  revol.util  inq.last.6mths
delinq.2yrs  \
0        5639.958333      28854        52.1               0
0
1        2760.000000      33623        76.7               0
0
2        4710.000000       3511        25.6               1
0
3        2699.958333      33667        73.2               1
0
4        4066.000000       4740        39.5               0
1

   pub.rec  not.fully.paid  purpose_all_other  purpose_credit_card  \
0        0               0                  0                    0
1        0               0                  0                    1
2        0               0                  0                    0
3        0               0                  0                    0
4        0               0                  0                    1

   purpose_debt_consolidation  purpose_educational
purpose_home_improvement  \
0                           1                    0
```

```
0  
1                            0                    0   
0  
2                            1                    0   
0  
3                            1                    0   
0  
4                            0                    0   
0  

   purpose_major_purchase  purpose_small_business  
0                       0                       0  
1                       0                       0  
2                       0                       0  
3                       0                       0  
4                       0                       0  
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 20 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   credit.policy               9578 non-null   int64  
 1   int.rate                    9578 non-null   float64
 2   installment                 9578 non-null   float64
 3   log.annual.inc              9578 non-null   float64
 4   dti                         9578 non-null   float64
 5   fico                        9578 non-null   int64  
 6   days.with.cr.line           9578 non-null   float64
 7   revol.bal                   9578 non-null   int64  
 8   revol.util                  9578 non-null   float64
 9   inq.last.6mths              9578 non-null   int64  
 10  delinq.2yrs                 9578 non-null   int64  
 11  pub.rec                     9578 non-null   int64  
 12  not.fully.paid              9578 non-null   int64  
 13  purpose_all_other           9578 non-null   uint8  
 14  purpose_credit_card         9578 non-null   uint8  
 15  purpose_debt_consolidation  9578 non-null   uint8  
 16  purpose_educational         9578 non-null   uint8  
 17  purpose_home_improvement    9578 non-null   uint8  
 18  purpose_major_purchase      9578 non-null   uint8  
 19  purpose_small_business      9578 non-null   uint8  
dtypes: float64(6), int64(7), uint8(7)
memory usage: 1.0 MB
```

EDA

```
data.describe().T
```

```
                              count          mean           std
min  \
credit.policy                9578.0      0.804970      0.396245
0.000000
int.rate                     9578.0      0.122640      0.026847
0.060000
installment                  9578.0    319.089413    207.071301
15.670000
log.annual.inc               9578.0     10.932117      0.614813
7.547502
dti                          9578.0     12.606679      6.883970
0.000000
fico                         9578.0    710.846314     37.970537
612.000000
days.with.cr.line            9578.0   4560.767197   2496.930377
178.958333
revol.bal                    9578.0  16913.963876  33756.189557
0.000000
revol.util                   9578.0     46.799236     29.014417
0.000000
inq.last.6mths               9578.0      1.577469      2.200245
0.000000
delinq.2yrs                  9578.0      0.163708      0.546215
0.000000
pub.rec                      9578.0      0.062122      0.262126
0.000000
not.fully.paid               9578.0      0.160054      0.366676
0.000000
purpose_all_other            9578.0      0.243370      0.429139
0.000000
purpose_credit_card          9578.0      0.131760      0.338248
0.000000
purpose_debt_consolidation  9578.0      0.413134      0.492422
0.000000
purpose_educational          9578.0      0.035811      0.185829
0.000000
purpose_home_improvement     9578.0      0.065671      0.247720
0.000000
purpose_major_purchase       9578.0      0.045625      0.208682
0.000000
purpose_small_business       9578.0      0.064627      0.245880
0.000000

                                   25%          50%           75%  \
credit.policy                 1.000000     1.000000      1.000000
int.rate                      0.103900     0.122100      0.140700
installment                 163.770000   268.950000    432.762500
log.annual.inc               10.558414    10.928884     11.291293
dti                           7.212500    12.665000     17.950000
fico                        682.000000   707.000000    737.000000
```

```
days.with.cr.line           2820.000000  4139.958333   5730.000000
revol.bal                   3187.000000  8596.000000  18249.500000
revol.util                    22.600000    46.300000     70.900000
inq.last.6mths                 0.000000     1.000000      2.000000
delinq.2yrs                    0.000000     0.000000      0.000000
pub.rec                        0.000000     0.000000      0.000000
not.fully.paid                 0.000000     0.000000      0.000000
purpose_all_other              0.000000     0.000000      0.000000
purpose_credit_card            0.000000     0.000000      0.000000
purpose_debt_consolidation     0.000000     0.000000      1.000000
purpose_educational            0.000000     0.000000      0.000000
purpose_home_improvement       0.000000     0.000000      0.000000
purpose_major_purchase         0.000000     0.000000      0.000000
purpose_small_business         0.000000     0.000000      0.000000

                                     max
credit.policy               1.000000e+00
int.rate                    2.164000e-01
installment                 9.401400e+02
log.annual.inc              1.452835e+01
dti                         2.996000e+01
fico                        8.270000e+02
days.with.cr.line           1.763996e+04
revol.bal                   1.207359e+06
revol.util                  1.190000e+02
inq.last.6mths              3.300000e+01
delinq.2yrs                 1.300000e+01
pub.rec                     5.000000e+00
not.fully.paid              1.000000e+00
purpose_all_other           1.000000e+00
purpose_credit_card         1.000000e+00
purpose_debt_consolidation  1.000000e+00
purpose_educational         1.000000e+00
purpose_home_improvement    1.000000e+00
purpose_major_purchase      1.000000e+00
purpose_small_business      1.000000e+00
```

```
data.corr()
```

```
                            credit.policy  int.rate  installment  \
credit.policy                    1.000000 -0.294089     0.058770
int.rate                        -0.294089  1.000000     0.276140
installment                      0.058770  0.276140     1.000000
log.annual.inc                   0.034906  0.056383     0.448102
dti                             -0.090901  0.220006     0.050202
fico                             0.348319 -0.714821     0.086039
days.with.cr.line                0.099026 -0.124022     0.183297
revol.bal                       -0.187518  0.092527     0.233625
revol.util                      -0.104095  0.464837     0.081356
inq.last.6mths                  -0.535511  0.202780    -0.010419
```

delinq.2yrs	-0.076318	0.156079	-0.004368
pub.rec	-0.054243	0.098162	-0.032760
not.fully.paid	-0.158119	0.159552	0.049955
purpose_all_other	-0.025412	-0.124000	-0.203103
purpose_credit_card	0.003216	-0.042109	0.000774
purpose_debt_consolidation	0.020193	0.123607	0.161658
purpose_educational	-0.031346	-0.019618	-0.094510
purpose_home_improvement	0.006036	-0.050697	0.023024
purpose_major_purchase	0.024281	-0.068978	-0.079836
purpose_small_business	-0.003511	0.151247	0.145654

	log.annual.inc	dti	fico \
credit.policy	0.034906	-0.090901	0.348319
int.rate	0.056383	0.220006	-0.714821
installment	0.448102	0.050202	0.086039
log.annual.inc	1.000000	-0.054065	0.114576
dti	-0.054065	1.000000	-0.241191
fico	0.114576	-0.241191	1.000000
days.with.cr.line	0.336896	0.060101	0.263880
revol.bal	0.372140	0.188748	-0.015553
revol.util	0.054881	0.337109	-0.541289
inq.last.6mths	0.029171	0.029189	-0.185293
delinq.2yrs	0.029203	-0.021792	-0.216340
pub.rec	0.016506	0.006209	-0.147592
not.fully.paid	-0.033439	0.037362	-0.149666
purpose_all_other	-0.080077	-0.125825	0.067184
purpose_credit_card	0.072942	0.084476	-0.012512
purpose_debt_consolidation	-0.026214	0.179149	-0.154132
purpose_educational	-0.119799	-0.035325	-0.013012
purpose_home_improvement	0.116375	-0.092788	0.097474
purpose_major_purchase	-0.031020	-0.077719	0.067129
purpose_small_business	0.091540	-0.069245	0.063292

	days.with.cr.line	revol.bal	revol.util \
credit.policy	0.099026	-0.187518	-0.104095
int.rate	-0.124022	0.092527	0.464837
installment	0.183297	0.233625	0.081356
log.annual.inc	0.336896	0.372140	0.054881
dti	0.060101	0.188748	0.337109
fico	0.263880	-0.015553	-0.541289
days.with.cr.line	1.000000	0.229344	-0.024239

revol.bal	0.229344	1.000000	0.203779
revol.util	-0.024239	0.203779	1.000000
inq.last.6mths	-0.041736	0.022394	-0.013880
delinq.2yrs	0.081374	-0.033243	-0.042740
pub.rec	0.071826	-0.031010	0.066717
not.fully.paid	-0.029237	0.053699	0.082088
purpose_all_other	-0.056574	-0.067728	-0.138535
purpose_credit_card	0.046220	0.072316	0.091321
purpose_debt_consolidation	-0.009318	0.005785	0.211869
purpose_educational	-0.042621	-0.034743	-0.053128
purpose_home_improvement	0.068087	0.003258	-0.114449
purpose_major_purchase	-0.020561	-0.062395	-0.108079
purpose_small_business	0.034883	0.083069	-0.060962

	inq.last.6mths	delinq.2yrs	pub.rec	\
credit.policy	-0.535511	-0.076318	-0.054243	
int.rate	0.202780	0.156079	0.098162	
installment	-0.010419	-0.004368	-0.032760	
log.annual.inc	0.029171	0.029203	0.016506	
dti	0.029189	-0.021792	0.006209	
fico	-0.185293	-0.216340	-0.147592	
days.with.cr.line	-0.041736	0.081374	0.071826	
revol.bal	0.022394	-0.033243	-0.031010	
revol.util	-0.013880	-0.042740	0.066717	
inq.last.6mths	1.000000	0.021245	0.072673	
delinq.2yrs	0.021245	1.000000	0.009184	
pub.rec	0.072673	0.009184	1.000000	
not.fully.paid	0.149452	0.008881	0.048634	
purpose_all_other	0.017795	0.016658	-0.030451	
purpose_credit_card	-0.033640	-0.008817	0.014842	
purpose_debt_consolidation	-0.044240	-0.000697	0.026845	
purpose_educational	0.024243	-0.002214	-0.013521	
purpose_home_improvement	0.043827	-0.013098	0.004704	
purpose_major_purchase	-0.001445	0.004085	-0.011734	
purpose_small_business	0.042567	-0.004148	-0.005595	

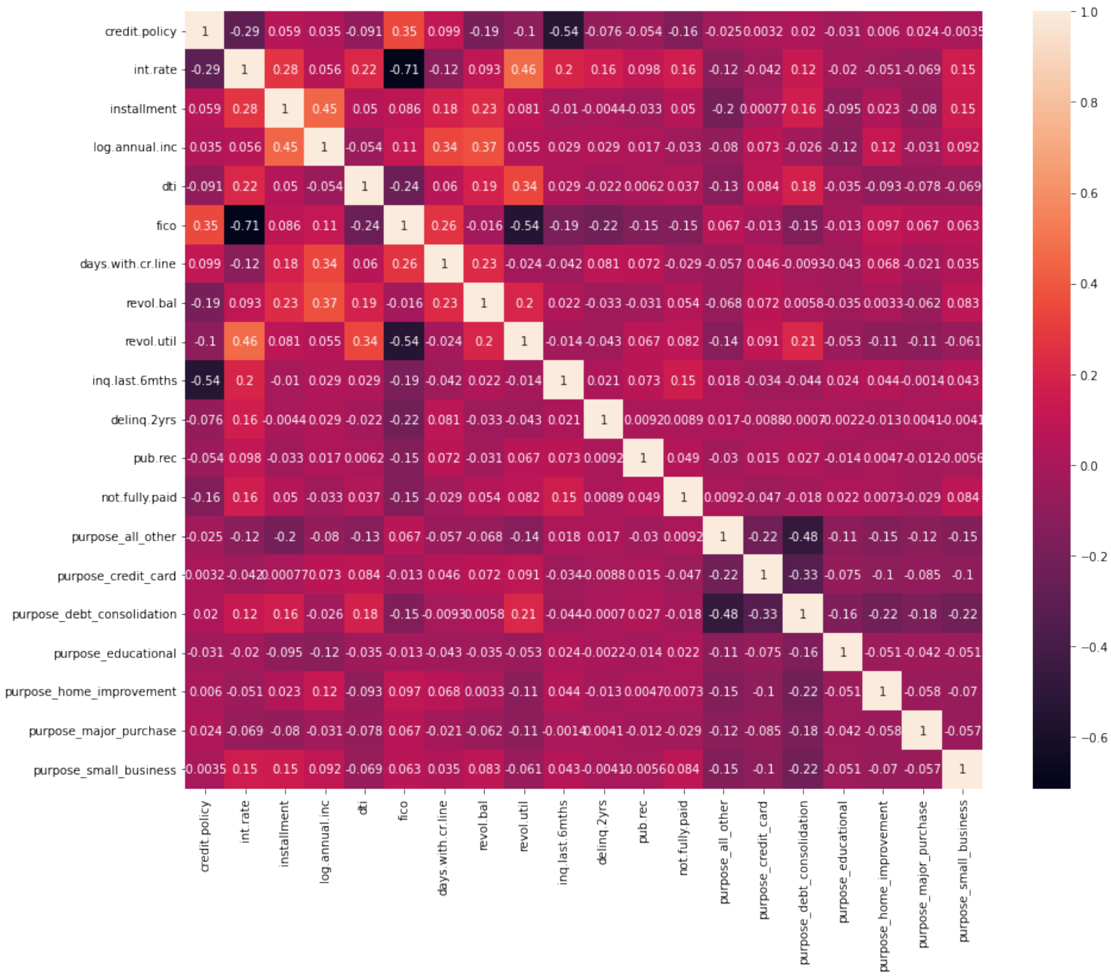
```
                           not.fully.paid  purpose_all_other  \
credit.policy                   -0.158119          -0.025412   
int.rate                         0.159552          -0.124000   
installment                      0.049955          -0.203103   
log.annual.inc                  -0.033439          -0.080077   
dti                              0.037362          -0.125825   
fico                            -0.149666           0.067184   
days.with.cr.line               -0.029237          -0.056574   
revol.bal                        0.053699          -0.067728   
revol.util                       0.082088          -0.138535   
inq.last.6mths                   0.149452           0.017795   
delinq.2yrs                      0.008881           0.016658   
pub.rec                          0.048634          -0.030451   
not.fully.paid                   1.000000           0.009233   
purpose_all_other                0.009233           1.000000   
purpose_credit_card             -0.047136          -0.220935   
purpose_debt_consolidation      -0.017543          -0.475848   
purpose_educational              0.021609          -0.109300   
purpose_home_improvement         0.007272          -0.150359   
purpose_major_purchase          -0.028580          -0.124004   
purpose_small_business           0.084460          -0.149076   

                           purpose_credit_card  
purpose_debt_consolidation  \
credit.policy                         0.003216                     
0.020193   
int.rate                             -0.042109                     
0.123607   
installment                           0.000774                     
0.161658   
log.annual.inc                        0.072942                   -
0.026214   
dti                                   0.084476                     
0.179149   
fico                                 -0.012512                   -
0.154132   
days.with.cr.line                     0.046220                   -
0.009318   
revol.bal                             0.072316                     
0.005785   
revol.util                            0.091321                     
0.211869   
inq.last.6mths                       -0.033640                   -
0.044240   
delinq.2yrs                          -0.008817                   -
0.000697   
pub.rec                               0.014842                     
0.026845   
not.fully.paid                       -0.047136                   -
0.017543   
```

```
purpose_all_other                    -0.220935                  -
0.475848
purpose_credit_card                   1.000000                  -
0.326850
purpose_debt_consolidation           -0.326850
1.000000
purpose_educational                  -0.075076                  -
0.161698
purpose_home_improvement             -0.103279                  -
0.222441
purpose_major_purchase               -0.085176                  -
0.183451
purpose_small_business               -0.102397                  -
0.220542

                            purpose_educational
purpose_home_improvement  \
credit.policy                        -0.031346
0.006036
int.rate                             -0.019618                 -
0.050697
installment                          -0.094510
0.023024
log.annual.inc                       -0.119799
0.116375
dti                                  -0.035325                 -
0.092788
fico                                 -0.013012
0.097474
days.with.cr.line                    -0.042621
0.068087
revol.bal                            -0.034743
0.003258
revol.util                           -0.053128                 -
0.114449
inq.last.6mths                        0.024243
0.043827
delinq.2yrs                          -0.002214                 -
0.013098
pub.rec                              -0.013521
0.004704
not.fully.paid                        0.021609
0.007272
purpose_all_other                    -0.109300                 -
0.150359
purpose_credit_card                  -0.075076                 -
0.103279
purpose_debt_consolidation           -0.161698                 -
0.222441
purpose_educational                   1.000000                 -
```

```
0.051094
purpose_home_improvement                   -0.051094
1.000000
purpose_major_purchase                     -0.042138                  -
0.057967
purpose_small_business                     -0.050658                  -
0.069687

                            purpose_major_purchase
purpose_small_business
credit.policy                                  0.024281                  -
0.003511
int.rate                                      -0.068978
0.151247
installment                                   -0.079836
0.145654
log.annual.inc                                -0.031020
0.091540
dti                                           -0.077719                  -
0.069245
fico                                           0.067129
0.063292
days.with.cr.line                             -0.020561
0.034883
revol.bal                                     -0.062395
0.083069
revol.util                                    -0.108079                  -
0.060962
inq.last.6mths                                -0.001445
0.042567
delinq.2yrs                                    0.004085                  -
0.004148
pub.rec                                       -0.011734                  -
0.005595
not.fully.paid                                -0.028580
0.084460
purpose_all_other                             -0.124004                  -
0.149076
purpose_credit_card                           -0.085176                  -
0.102397
purpose_debt_consolidation                    -0.183451                  -
0.220542
purpose_educational                           -0.042138                  -
0.050658
purpose_home_improvement                      -0.057967                  -
0.069687
purpose_major_purchase                         1.000000                  -
0.057472
purpose_small_business                        -0.057472
1.000000
```

Corelation Heatmap before splitting

```
plt.figure(figsize = (15,12))
sns.heatmap(data = data.corr(), annot = True)
plt.show()
```

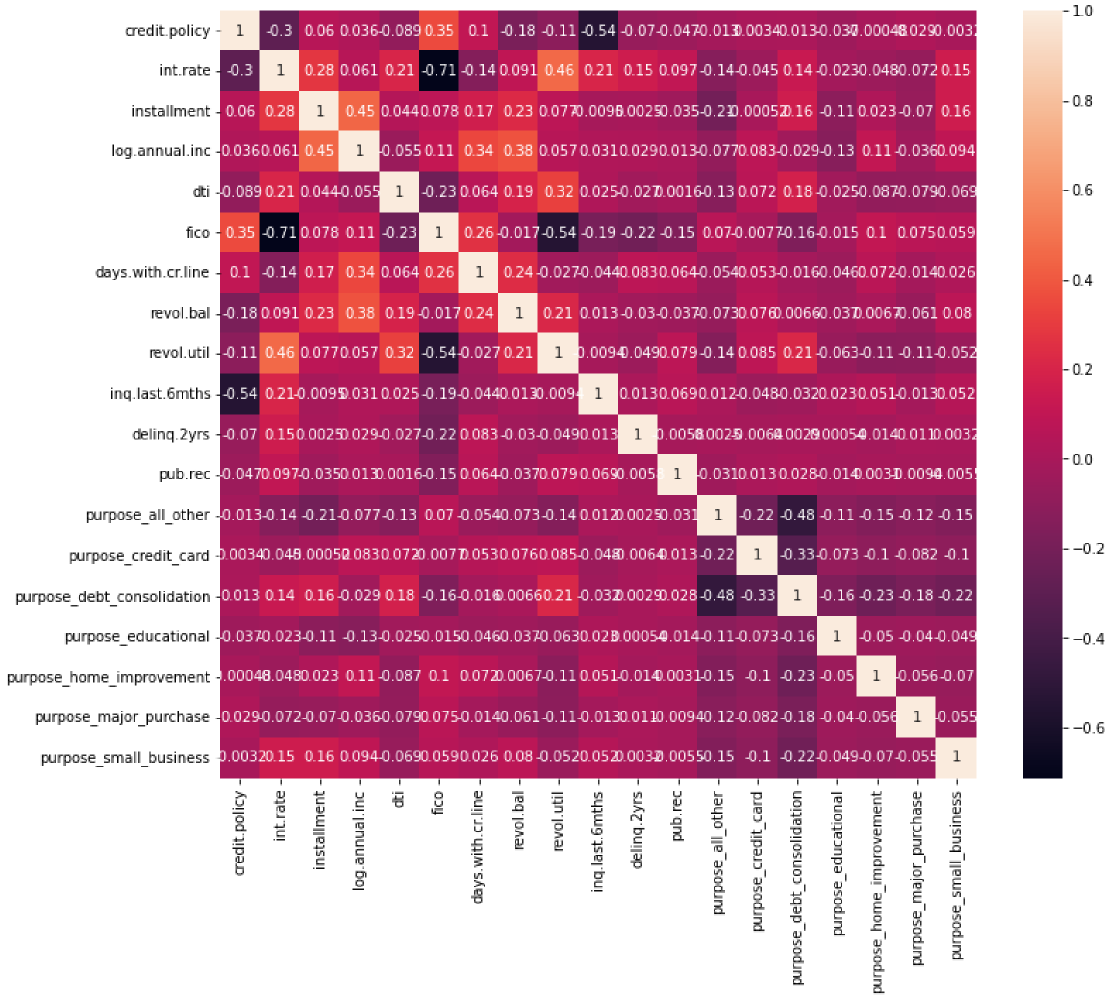


Splitting data

```
X = data.drop("not.fully.paid", axis = 1)
y = data["not.fully.paid"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.3, random_state = 4)
```

Corelation Heatmap after data split

```
plt.figure(figsize = (12,10))
sns.heatmap(data = X_train.corr(), annot = True)
plt.show()
```



from the correlation heatmaps we can observe that no two features have positive corelation of more than 0.7 , so we will not remove any feature.

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
print(X_train.shape , X_test.shape)
```

```
(6704, 19) (2874, 19)
```

Building Deep Learning Model and Defining Model architecture

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, BatchNormalization, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
```

```
model = Sequential()

model.add(Dense(units = 128, activation = 'relu', input_shape =
(X_train.shape[1],)))
model.add(BatchNormalization())
model.add(Dropout(0.2, seed = 123))

model.add(Dense(units = 64, activation = 'tanh'))
model.add(BatchNormalization())
model.add(Dropout(0.2, seed = 123))

model.add(Dense(units = 32, activation = 'relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2, seed = 123))

model.add(Dense(units = 1, activation = 'sigmoid'))

es = EarlyStopping(monitor = "accuracy", patience = 4)

model.compile(optimizer = Adam(learning_rate = 0.01),
 loss = 'binary_crossentropy',
 metrics = ['accuracy'])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               2560

 batch_normalization (BatchN  (None, 128)              512
 ormalization)

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 batch_normalization_1 (Batc  (None, 64)               256
 hNormalization)

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 32)                2080

 batch_normalization_2 (Batc  (None, 32)               128
 hNormalization)

 dropout_2 (Dropout)         (None, 32)                0

 dense_3 (Dense)             (None, 1)                 33
```

```
=================================================================
Total params: 13,825
Trainable params: 13,377
Non-trainable params: 448
_________________________________________________________________
```

Model Training

```
result = model.fit(X_train, y_train,
 validation_data = (X_test, y_test),
 callbacks = [es],
 epochs = 100)
```

```
Epoch 1/100
210/210 [==============================] - 3s 5ms/step - loss: 0.4711
- accuracy: 0.8125 - val_loss: 0.4075 - val_accuracy: 0.8497
Epoch 2/100
210/210 [==============================] - 1s 4ms/step - loss: 0.4280
- accuracy: 0.8358 - val_loss: 0.4000 - val_accuracy: 0.8497
Epoch 3/100
210/210 [==============================] - 1s 4ms/step - loss: 0.4211
- accuracy: 0.8355 - val_loss: 0.3985 - val_accuracy: 0.8497
Epoch 4/100
210/210 [==============================] - 1s 4ms/step - loss: 0.4209
- accuracy: 0.8349 - val_loss: 0.3973 - val_accuracy: 0.8497
Epoch 5/100
210/210 [==============================] - 1s 4ms/step - loss: 0.4210
- accuracy: 0.8355 - val_loss: 0.3980 - val_accuracy: 0.8497
Epoch 6/100
210/210 [==============================] - 1s 4ms/step - loss: 0.4203
- accuracy: 0.8358 - val_loss: 0.3995 - val_accuracy: 0.8497
```

```
y_pred = model.predict(X_test) >=0.5
```

```
90/90 [==============================] - 0s 2ms/step
```

Model testing

```
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy_score(y_pred, y_test)
```

```
0.8496868475991649
```

```
confusion_matrix(y_pred, y_test)
```

```
array([[2442,  432],
       [   0,    0]])
```

Hyperparameter Tunning

```
!pip install -q -U keras-tuner
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.1/168.1 KB 5.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 41.1 MB/s eta
0:00:00
```

defining limits of parameters for tuning

```python
def build_model(hp):
 model = Sequential()

 model.add(Dense(units = hp.Int('units', min_value = 32, max_value = 
1024, step = 16),
                                 activation = hp.Choice('actiivation', 
['relu','tanh']),
                                 input_shape = (X_train.shape[1],)))
 model.add(BatchNormalization())
 model.add(Dropout(hp.Float('rate', min_value = 0.1, max_value = 0.4, 
step = 0.1), seed = 1234))

 model.add(Dense(units = hp.Int("units", min_value = 32, max_value = 
128, step = 16),
                                 activation = hp.Choice("activation", 
["relu","tanh"])))
 model.add(BatchNormalization())
 model.add(Dropout(hp.Float("rate",min_value = 0.1, max_value = 0.4, 
step = 0.1), seed = 1234))

 model.add(Dense(units = hp.Int("units", min_value = 32, max_value = 
64, step = 16),
                                 activation = hp.Choice("activation", 
["relu","tanh"])))
 model.add(BatchNormalization())
 model.add(Dropout(hp.Float("rate",min_value = 0.1, max_value = 0.4, 
step = 0.1), seed = 1234))

 model.add(Dense(units = 1, activation = "sigmoid"))

 learning_rate = hp.Float("learning_rate", min_value = 0.001, 
max_value = 0.1, step = 0.01)

 model.compile(optimizer = tf.keras.optimizers.Adam(learning_rate),
               loss = "binary_crossentropy",
               metrics = ["accuracy"])

 return model

import keras_tuner as kt
build_model(kt.HyperParameters())
```

```
<keras.engine.sequential.Sequential at 0x7ff11c568460>
```

Setting rules for tuning

```
rtuner = kt.RandomSearch(hypermodel = build_model,
 objective='val_accuracy',
 max_trials=3,
 executions_per_trial=2,
 overwrite=True,
 directory='my_dir',
 project_name='diab')
```

Tuning the parameters

```
rtuner.search(X_train, y_train, epochs=2, validation_data=(X_test,
y_test))
```

```
Trial 3 Complete [00h 00m 12s]
val_accuracy: 0.849686861038208

Best val_accuracy So Far: 0.849686861038208
Total elapsed time: 00h 00m 50s
```

Chosing any one model, there are two model created namely models[0] and model[1]

```
models = rtuner.get_best_models(num_models=2)
models[1].summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 240)               4800

 batch_normalization (BatchN  (None, 240)              960
 ormalization)

 dropout (Dropout)           (None, 240)               0

 dense_1 (Dense)             (None, 240)               57840

 batch_normalization_1 (Batc  (None, 240)              960
 hNormalization)

 dropout_1 (Dropout)         (None, 240)               0

 dense_2 (Dense)             (None, 240)               57840

 batch_normalization_2 (Batc  (None, 240)              960
 hNormalization)

 dropout_2 (Dropout)         (None, 240)               0

 dense_3 (Dense)             (None, 1)                 241
```

```
=================================================================
Total params: 123,601
Trainable params: 122,161
Non-trainable params: 1,440
_________________________________________________________________
```

```
y_predh = models[1].predict(X_test) >=0.5
accuracy_score(y_predh, y_test)
```

```
90/90 [==============================] - 0s 2ms/step
```

```
0.8496868475991649
```