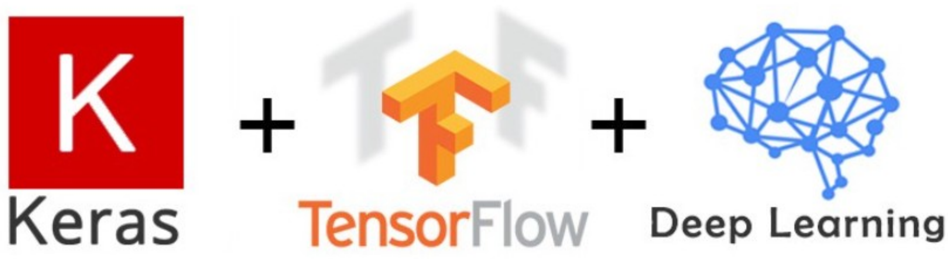
```
from IPython.display import Image
Image(filename='Capture.png')
```



Predicting Loan Defaults using Deep Learning with Keras & Tensorflow

import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential   #pip uninstall h5py
#conda install -c anaconda h5py
from tensorflow.keras.layers import Dense,Dropout
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.models import load_model
from sklearn.metrics import confusion_matrix, classification_report
from pickle import dump, load

%matplotlib inline

df = pd.read_csv(r"C:/Users/Nageswaran
B/Documents/machine_learning/machine_learning_project/simplilearn/
Lending_Club_Loan_Data_Analysis/loan_data.csv")
df.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
```

```
df.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
```

The “Purpose” data column is categorical, “Annual income” is log value, which needs to be converted back to exponential. The rest of the columns are numerical. Transpose the data frame to understand the std and mean.

```
df.describe().transpose()

                    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

                           25%          50%           75%
max
credit.policy         1.000000     1.000000      1.000000
1.000000e+00
int.rate              0.103900     0.122100      0.140700  2.164000e-
01
installment         163.770000   268.950000    432.762500
9.401400e+02
log.annual.inc       10.558414    10.928884     11.291293
1.452835e+01
dti                   7.212500    12.665000     17.950000
2.996000e+01
fico                682.000000   707.000000    737.000000
8.270000e+02
days.with.cr.line  2820.000000  4139.958333   5730.000000
1.763996e+04
revol.bal          3187.000000  8596.000000  18249.500000
1.207359e+06
revol.util           22.600000    46.300000     70.900000
1.190000e+02
inq.last.6mths        0.000000     1.000000      2.000000
3.300000e+01
delinq.2yrs           0.000000     0.000000      0.000000
1.300000e+01
pub.rec               0.000000     0.000000      0.000000
5.000000e+00
not.fully.paid        0.000000     0.000000      0.000000
1.000000e+00
```

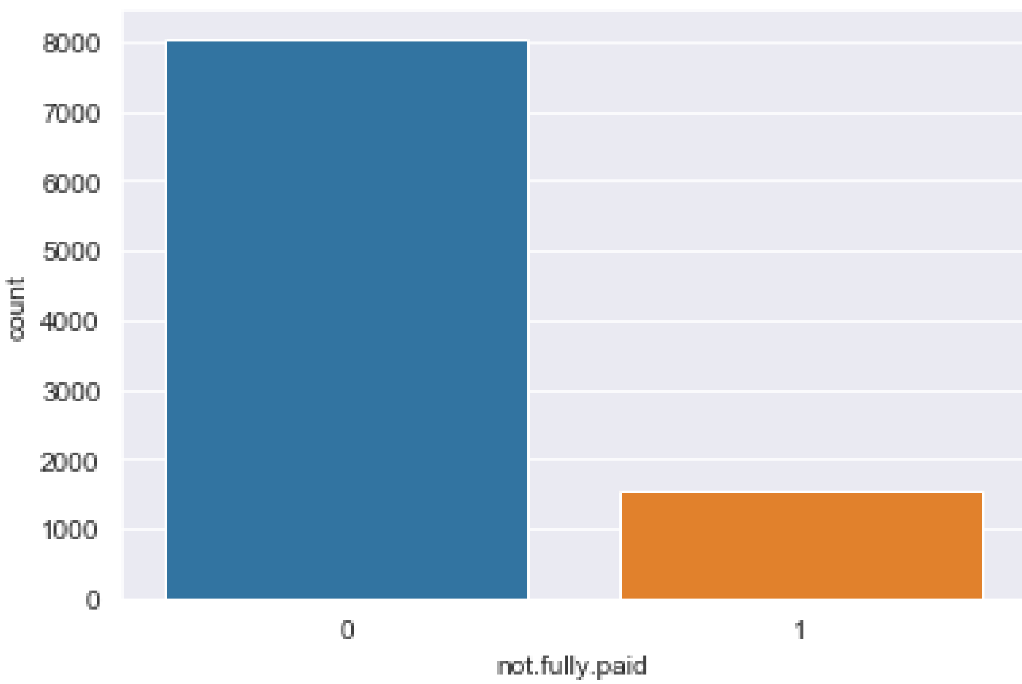
Check the label “no.fully.paid” distribution in the dataset.

```
df['not.fully.paid'].isnull().mean()
df.groupby('not.fully.paid')['not.fully.paid'].count()/len(df)
```

```
not.fully.paid
0    0.839946
1    0.160054
Name: not.fully.paid, dtype: float64
```

```
sns.set_style('darkgrid')
sns.countplot(x='not.fully.paid', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2a40177aa90>
```



The above shows, This dataset is highly imbalanced and includes features that make this problem more challenging. If we do model training with this data, the prediction will be biased since the “not.fully.paid =0 “ has 83.9% filled, and only 16% is the “not.fully.paid=1”

There were multiple methods to handle imbalanced data; here are a few techniques.

1. Resample the training set

There are two approaches to make a balanced dataset out of an imbalanced one are under-sampling and over-sampling.

Under-sampling

Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when the quantity of data is sufficient.

Over-sampling

Oversampling is used when the quantity of data is insufficient. It tries to balance the dataset by increasing the size of rare samples.

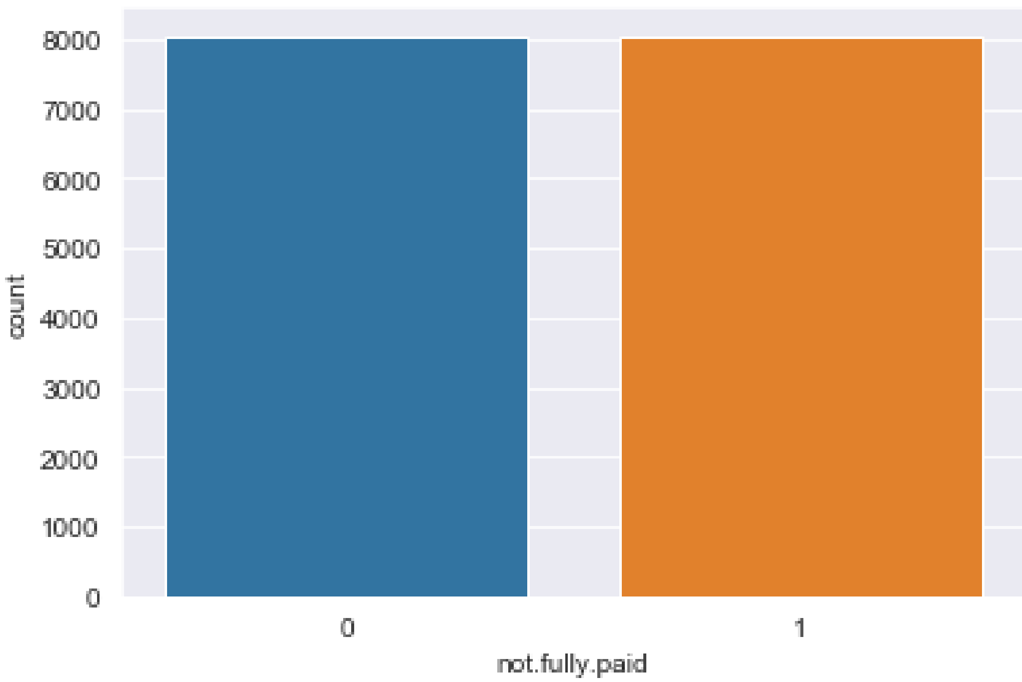
There is no absolute advantage of one resampling method over another.

```
count_class_0, count_class_1 = df['not.fully.paid'].value_counts()
df_0 = df[df['not.fully.paid'] == 0]
df_1 = df[df['not.fully.paid'] == 1]
df_1_over = df_1.sample(count_class_0, replace=True)
df_test_over = pd.concat([df_0, df_1_over], axis=0)
print('Random over-sampling:')
print(df_test_over['not.fully.paid'].value_counts())

sns.set_style('darkgrid')
sns.countplot(x='not.fully.paid', data=df_test_over)
```

```
Random over-sampling:
0    8045
1    8045
Name: not.fully.paid, dtype: int64
```

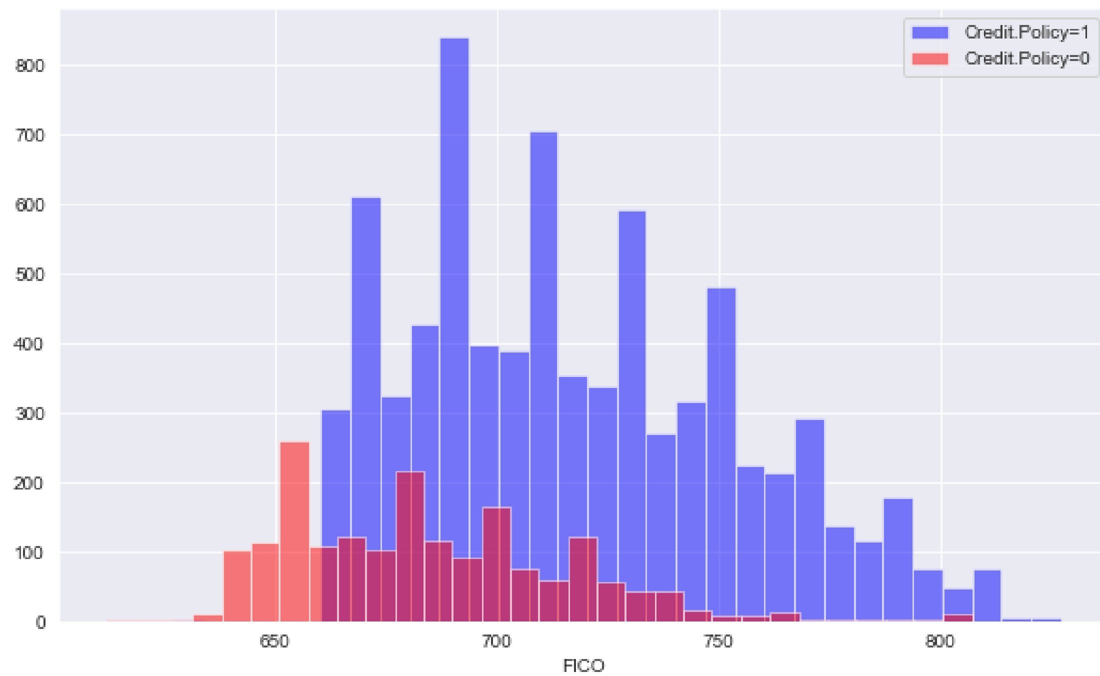
```
<matplotlib.axes._subplots.AxesSubplot at 0x2a401ad3e48>
```



Exploratory Data Analysis

```
plt.figure(figsize=(10,6))
df[df['credit.policy']==1]
['fico'].hist(alpha=0.5,color='blue',bins=30,label='Credit.Policy=1')
df[df['credit.policy']==0]
['fico'].hist(alpha=0.5,color='red',bins=30,label='Credit.Policy=0')
plt.legend()
plt.xlabel('FICO')
```

```
Text(0.5, 0, 'FICO')
```

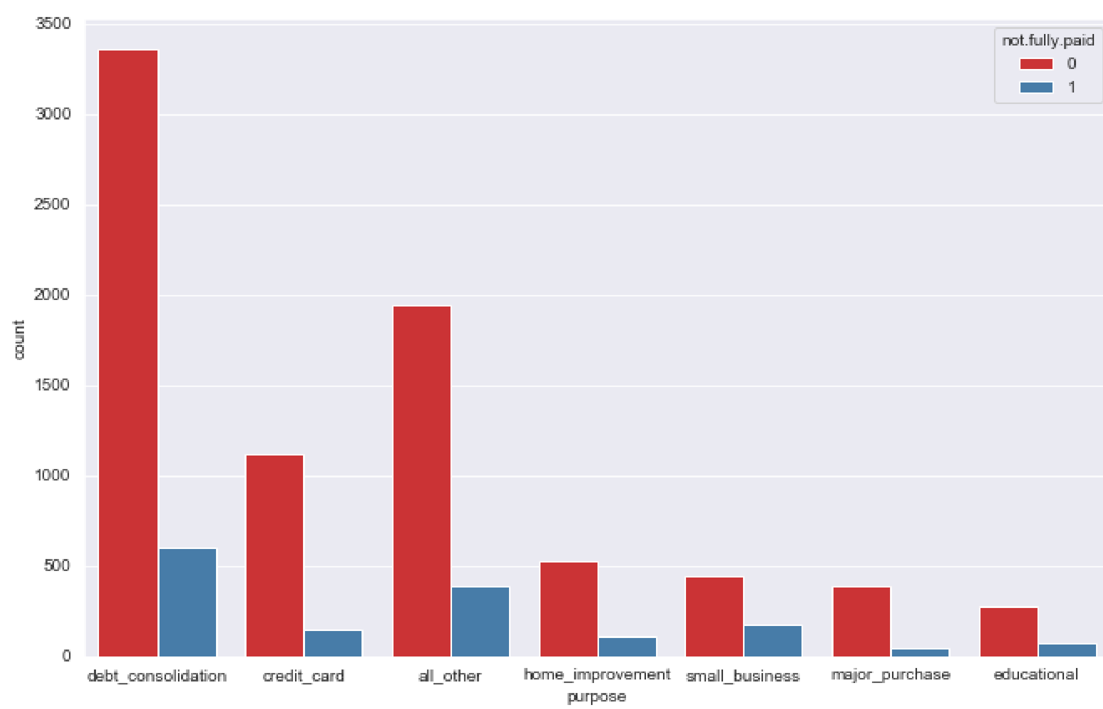



```
plt.figure(figsize=(10,6))
df[df['not.fully.paid']==1]
['fico'].hist(alpha=0.5,color='blue',bins=30,label='not.fully.paid=1')
df[df['not.fully.paid']==0]
['fico'].hist(alpha=0.5,color='red',bins=30,label='not.fully.paid=0')
plt.legend()
plt.xlabel('FICO')
```

```
Text(0.5, 0, 'FICO')
```

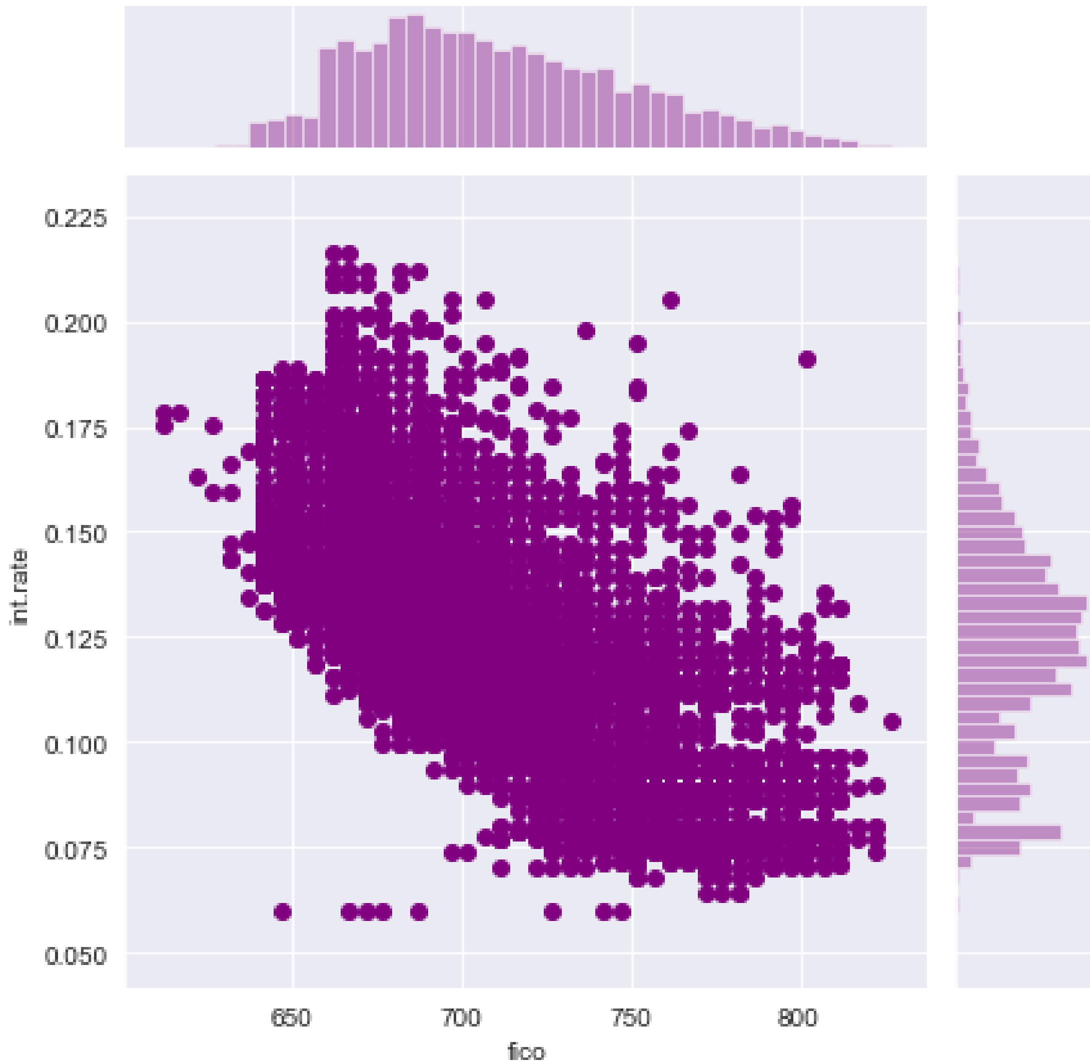

```
plt.figure(figsize=(11,7))
sns.countplot(x='purpose',hue='not.fully.paid',data=df,palette='Set1')
```

<matplotlib.axes._subplots.AxesSubplot at 0x2a402dc3940>




```
sns.jointplot(x='fico',y='int.rate',data=df,color='purple')
```

<seaborn.axisgrid.JointGrid at 0x2a402e13208>

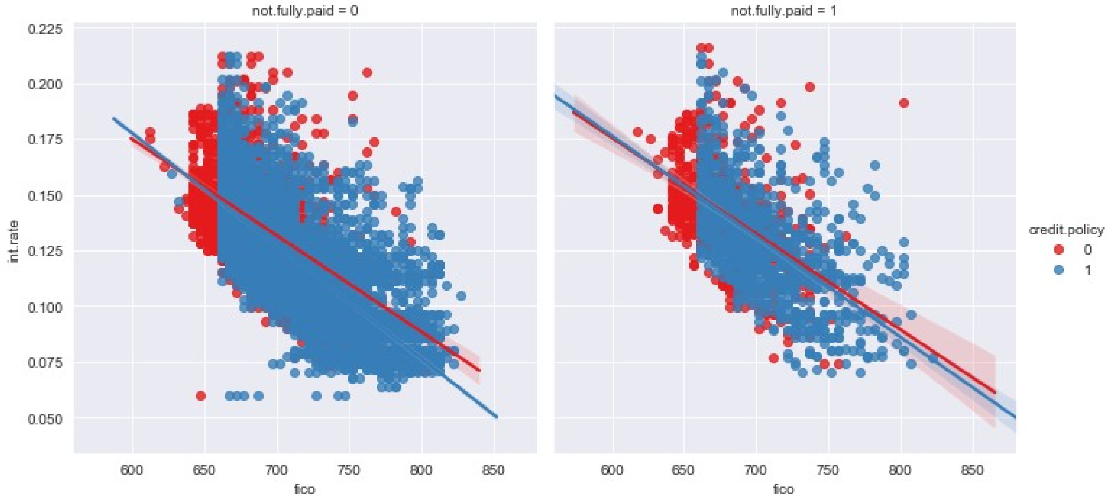


To compare the trend between not.fully.paid and credit.policy, create seaborn implot.

```
plt.figure(figsize=(11,7))
sns.lmplot(y='int.rate',x='fico',data=df,hue='credit.policy',
           col='not.fully.paid',palette='Set1')
```

```
<seaborn.axisgrid.FacetGrid at 0x2a4031a34e0>
```

```
<Figure size 792x504 with 0 Axes>
```



The above visuals gave us an idea of how the data is and what we will work with. Nest step is to prepare the data for model training and test as the first step converts the categorical values to numeric. Here in this dataset "purpose" column is a critical data point for the model as per our analysis above, and it is categorical.

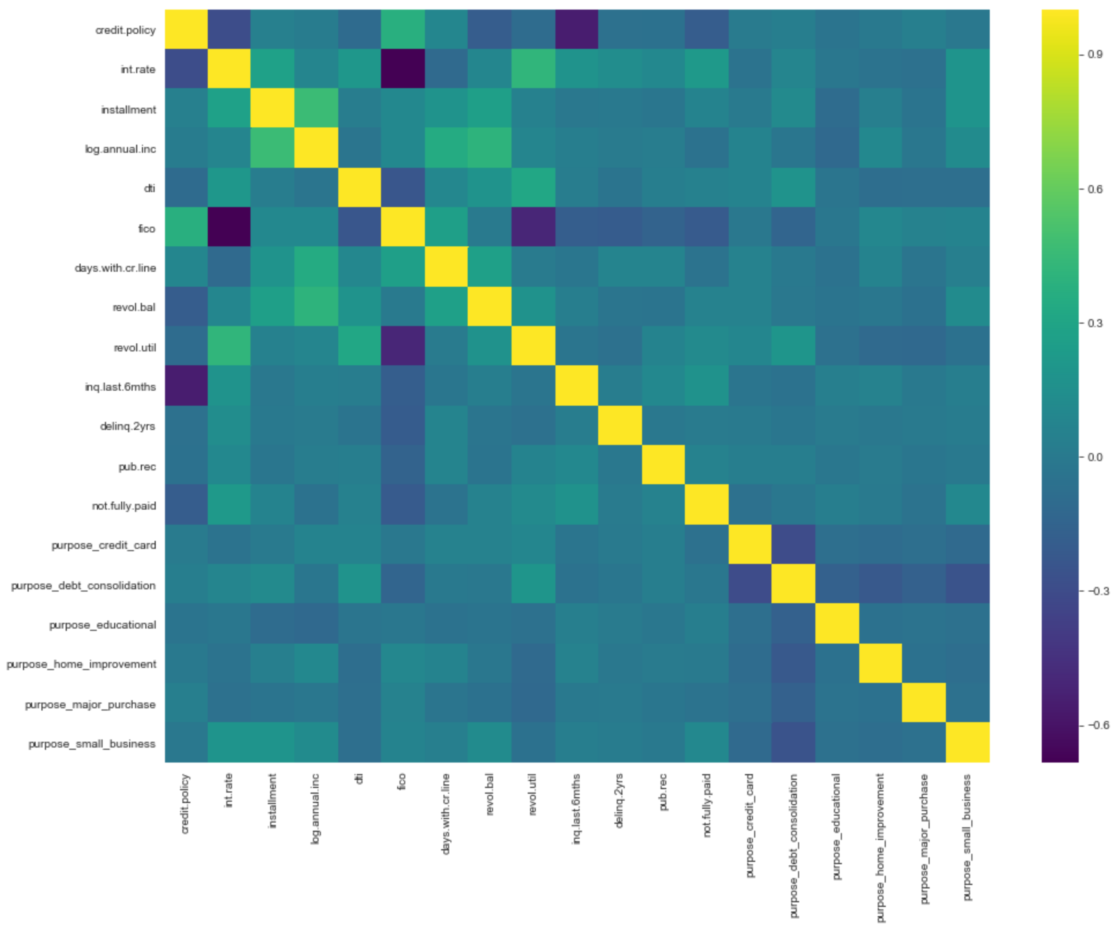
```
col_fea = ['purpose']
final_data =
pd.get_dummies(df_test_over,columns=col_fea,drop_first=True)
final_data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 16090 entries, 0 to 8012
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   credit.policy               16090 non-null  int64
 1   int.rate                    16090 non-null  float64
 2   installment                 16090 non-null  float64
 3   log.annual.inc              16090 non-null  float64
 4   dti                         16090 non-null  float64
 5   fico                        16090 non-null  int64
 6   days.with.cr.line           16090 non-null  float64
 7   revol.bal                   16090 non-null  int64
 8   revol.util                  16090 non-null  float64
 9   inq.last.6mths              16090 non-null  int64
 10  delinq.2yrs                 16090 non-null  int64
 11  pub.rec                     16090 non-null  int64
 12  not.fully.paid              16090 non-null  int64
 13  purpose_credit_card         16090 non-null  uint8
 14  purpose_debt_consolidation  16090 non-null  uint8
 15  purpose_educational         16090 non-null  uint8
 16  purpose_home_improvement    16090 non-null  uint8
 17  purpose_major_purchase      16090 non-null  uint8
 18  purpose_small_business      16090 non-null  uint8
```

```
dtypes: float64(6), int64(7), uint8(6)
memory usage: 2.1 MB
```

```
final_data.corr()
plt.figure(
        figsize=[16,12]
)
sns.heatmap(
        data=final_data.corr(),
        cmap='viridis',
        annot=False,
        fmt='.2g'
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2a403280ac8>
```



We only focus on the grids of yellow or very light green. After comparing with the feature description again, I decided to drop:'revol.bal', 'days.with.cr.line', 'installment', 'revol.bal'

revol.bal, day.with.cr.line, installment can represent by annual income. revol.util can represent by int.rate.

Modeling

Deep Learning Implementation

Finally, do the train test split and fit the model with the data shape we created above. since there are 19 features, I chose the first layer of the neural network with 19 nodes.

```
to_train = final_data[final_data['not.fully.paid'].isin([0,1])]
to_pred = final_data[final_data['not.fully.paid'] == 2]

X = to_train.drop('not.fully.paid', axis=1).values
y = to_train['not.fully.paid'].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state = 101)

scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = Sequential()

model.add(
        Dense(19, activation='relu')
)

model.add(
        Dense(10, activation='relu')
)

model.add(
        Dense(5, activation='relu')
)


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

model.compile(
        optimizer='adam',
        loss='binary_crossentropy',
```

```
        metrics=['accuracy']
)

early_stop = EarlyStopping(
        monitor='val_loss',
        mode='min',
        verbose=1,
        patience=25
)

model.fit(
        X_train,
        y_train,
        epochs=200,
        batch_size=256,
        validation_data=(X_test, y_test),
         callbacks=[early_stop]
)
```

```
Epoch 1/200
44/44 [==============================] - 1s 5ms/step - loss: 0.6907 -
accuracy: 0.5401 - val_loss: 0.6884 - val_accuracy: 0.5844
Epoch 2/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6830 -
accuracy: 0.6041 - val_loss: 0.6794 - val_accuracy: 0.6002
Epoch 3/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6708 -
accuracy: 0.6135 - val_loss: 0.6677 - val_accuracy: 0.6051
Epoch 4/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6580 -
accuracy: 0.6145 - val_loss: 0.6575 - val_accuracy: 0.6105
Epoch 5/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6491 -
accuracy: 0.6203 - val_loss: 0.6530 - val_accuracy: 0.6130
Epoch 6/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6446 -
accuracy: 0.6230 - val_loss: 0.6517 - val_accuracy: 0.6169
Epoch 7/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6424 -
accuracy: 0.6220 - val_loss: 0.6511 - val_accuracy: 0.6109
Epoch 8/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6407 -
accuracy: 0.6261 - val_loss: 0.6513 - val_accuracy: 0.6101
Epoch 9/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6393 -
accuracy: 0.6219 - val_loss: 0.6501 - val_accuracy: 0.6093
Epoch 10/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6381 -
accuracy: 0.6258 - val_loss: 0.6494 - val_accuracy: 0.6128
Epoch 11/200
```

```
44/44 [==============================] - 0s 1ms/step - loss: 0.6371 -
accuracy: 0.6259 - val_loss: 0.6483 - val_accuracy: 0.6145
Epoch 12/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6362 -
accuracy: 0.6275 - val_loss: 0.6489 - val_accuracy: 0.6122
Epoch 13/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6351 -
accuracy: 0.6274 - val_loss: 0.6479 - val_accuracy: 0.6153
Epoch 14/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6347 -
accuracy: 0.6293 - val_loss: 0.6472 - val_accuracy: 0.6163
Epoch 15/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6338 -
accuracy: 0.6311 - val_loss: 0.6485 - val_accuracy: 0.6085
Epoch 16/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6335 -
accuracy: 0.6313 - val_loss: 0.6454 - val_accuracy: 0.6182
Epoch 17/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6327 -
accuracy: 0.6306 - val_loss: 0.6459 - val_accuracy: 0.6178
Epoch 18/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6328 -
accuracy: 0.6298 - val_loss: 0.6458 - val_accuracy: 0.6182
Epoch 19/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6318 -
accuracy: 0.6295 - val_loss: 0.6448 - val_accuracy: 0.6169
Epoch 20/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6310 -
accuracy: 0.6320 - val_loss: 0.6446 - val_accuracy: 0.6184
Epoch 21/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6303 -
accuracy: 0.6298 - val_loss: 0.6439 - val_accuracy: 0.6165
Epoch 22/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6293 -
accuracy: 0.6346 - val_loss: 0.6439 - val_accuracy: 0.6165
Epoch 23/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6294 -
accuracy: 0.6359 - val_loss: 0.6429 - val_accuracy: 0.6194
Epoch 24/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6291 -
accuracy: 0.6346 - val_loss: 0.6439 - val_accuracy: 0.6213
Epoch 25/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6290 -
accuracy: 0.6326 - val_loss: 0.6433 - val_accuracy: 0.6203
Epoch 26/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6280 -
accuracy: 0.6355 - val_loss: 0.6426 - val_accuracy: 0.6169
Epoch 27/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6276 -
accuracy: 0.6337 - val_loss: 0.6423 - val_accuracy: 0.6269
```

```
Epoch 28/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6272 -
accuracy: 0.6370 - val_loss: 0.6428 - val_accuracy: 0.6273
Epoch 29/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6269 -
accuracy: 0.6348 - val_loss: 0.6467 - val_accuracy: 0.6194
Epoch 30/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6272 -
accuracy: 0.6339 - val_loss: 0.6415 - val_accuracy: 0.6230
Epoch 31/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6262 -
accuracy: 0.6352 - val_loss: 0.6411 - val_accuracy: 0.6232
Epoch 32/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6256 -
accuracy: 0.6367 - val_loss: 0.6415 - val_accuracy: 0.6238
Epoch 33/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6252 -
accuracy: 0.6364 - val_loss: 0.6408 - val_accuracy: 0.6209
Epoch 34/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6247 -
accuracy: 0.6365 - val_loss: 0.6405 - val_accuracy: 0.6225
Epoch 35/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6246 -
accuracy: 0.6366 - val_loss: 0.6404 - val_accuracy: 0.6184
Epoch 36/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6245 -
accuracy: 0.6382 - val_loss: 0.6410 - val_accuracy: 0.6236
Epoch 37/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6242 -
accuracy: 0.6361 - val_loss: 0.6395 - val_accuracy: 0.6219
Epoch 38/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6242 -
accuracy: 0.6362 - val_loss: 0.6400 - val_accuracy: 0.6265
Epoch 39/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6235 -
accuracy: 0.6369 - val_loss: 0.6395 - val_accuracy: 0.6246
Epoch 40/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6236 -
accuracy: 0.6380 - val_loss: 0.6396 - val_accuracy: 0.6184
Epoch 41/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6229 -
accuracy: 0.6384 - val_loss: 0.6397 - val_accuracy: 0.6242
Epoch 42/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6225 -
accuracy: 0.6369 - val_loss: 0.6395 - val_accuracy: 0.6248
Epoch 43/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6219 -
accuracy: 0.6401 - val_loss: 0.6390 - val_accuracy: 0.6271
Epoch 44/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6215 -
```

```
accuracy: 0.6389 - val_loss: 0.6399 - val_accuracy: 0.6256
Epoch 45/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6218 -
accuracy: 0.6430 - val_loss: 0.6397 - val_accuracy: 0.6269
Epoch 46/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6209 -
accuracy: 0.6412 - val_loss: 0.6384 - val_accuracy: 0.6254
Epoch 47/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6205 -
accuracy: 0.6396 - val_loss: 0.6399 - val_accuracy: 0.6269
Epoch 48/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6205 -
accuracy: 0.6407 - val_loss: 0.6388 - val_accuracy: 0.6236
Epoch 49/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6201 -
accuracy: 0.6400 - val_loss: 0.6417 - val_accuracy: 0.6238
Epoch 50/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6222 -
accuracy: 0.6375 - val_loss: 0.6383 - val_accuracy: 0.6298
Epoch 51/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6200 -
accuracy: 0.6382 - val_loss: 0.6382 - val_accuracy: 0.6292
Epoch 52/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6214 -
accuracy: 0.6401 - val_loss: 0.6388 - val_accuracy: 0.6265
Epoch 53/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6191 -
accuracy: 0.6442 - val_loss: 0.6382 - val_accuracy: 0.6265
Epoch 54/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6188 -
accuracy: 0.6412 - val_loss: 0.6403 - val_accuracy: 0.6275
Epoch 55/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6189 -
accuracy: 0.6426 - val_loss: 0.6374 - val_accuracy: 0.6275
Epoch 56/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6185 -
accuracy: 0.6428 - val_loss: 0.6379 - val_accuracy: 0.6319
Epoch 57/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6183 -
accuracy: 0.6406 - val_loss: 0.6382 - val_accuracy: 0.6304
Epoch 58/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6179 -
accuracy: 0.6420 - val_loss: 0.6372 - val_accuracy: 0.6288
Epoch 59/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6182 -
accuracy: 0.6414 - val_loss: 0.6368 - val_accuracy: 0.6281
Epoch 60/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6173 -
accuracy: 0.6458 - val_loss: 0.6374 - val_accuracy: 0.6292
Epoch 61/200
```

```
44/44 [==============================] - 0s 1ms/step - loss: 0.6168 -
accuracy: 0.6444 - val_loss: 0.6378 - val_accuracy: 0.6294
Epoch 62/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6171 -
accuracy: 0.6422 - val_loss: 0.6364 - val_accuracy: 0.6256
Epoch 63/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6161 -
accuracy: 0.6463 - val_loss: 0.6370 - val_accuracy: 0.6302
Epoch 64/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6163 -
accuracy: 0.6436 - val_loss: 0.6382 - val_accuracy: 0.6294
Epoch 65/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6160 -
accuracy: 0.6463 - val_loss: 0.6372 - val_accuracy: 0.6310
Epoch 66/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6168 -
accuracy: 0.6449 - val_loss: 0.6393 - val_accuracy: 0.6327
Epoch 67/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6155 -
accuracy: 0.6444 - val_loss: 0.6362 - val_accuracy: 0.6277
Epoch 68/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6154 -
accuracy: 0.6463 - val_loss: 0.6357 - val_accuracy: 0.6308
Epoch 69/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6154 -
accuracy: 0.6449 - val_loss: 0.6364 - val_accuracy: 0.6271
Epoch 70/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6151 -
accuracy: 0.6454 - val_loss: 0.6374 - val_accuracy: 0.6312
Epoch 71/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6149 -
accuracy: 0.6455 - val_loss: 0.6354 - val_accuracy: 0.6279
Epoch 72/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6157 -
accuracy: 0.6472 - val_loss: 0.6361 - val_accuracy: 0.6246
Epoch 73/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6141 -
accuracy: 0.6463 - val_loss: 0.6368 - val_accuracy: 0.6290
Epoch 74/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6144 -
accuracy: 0.6437 - val_loss: 0.6372 - val_accuracy: 0.6337
Epoch 75/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6141 -
accuracy: 0.6456 - val_loss: 0.6353 - val_accuracy: 0.6250
Epoch 76/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6134 -
accuracy: 0.6476 - val_loss: 0.6352 - val_accuracy: 0.6298
Epoch 77/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6131 -
accuracy: 0.6484 - val_loss: 0.6348 - val_accuracy: 0.6298
```

```
Epoch 78/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6129 -
accuracy: 0.6466 - val_loss: 0.6379 - val_accuracy: 0.6300
Epoch 79/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6128 -
accuracy: 0.6471 - val_loss: 0.6398 - val_accuracy: 0.6285
Epoch 80/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6135 -
accuracy: 0.6476 - val_loss: 0.6360 - val_accuracy: 0.6327
Epoch 81/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6121 -
accuracy: 0.6477 - val_loss: 0.6349 - val_accuracy: 0.6341
Epoch 82/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6119 -
accuracy: 0.6489 - val_loss: 0.6353 - val_accuracy: 0.6308
Epoch 83/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6119 -
accuracy: 0.6493 - val_loss: 0.6354 - val_accuracy: 0.6310
Epoch 84/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6113 -
accuracy: 0.6514 - val_loss: 0.6358 - val_accuracy: 0.6312
Epoch 85/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6110 -
accuracy: 0.6478 - val_loss: 0.6347 - val_accuracy: 0.6337
Epoch 86/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6116 -
accuracy: 0.6459 - val_loss: 0.6378 - val_accuracy: 0.6323
Epoch 87/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6105 -
accuracy: 0.6517 - val_loss: 0.6355 - val_accuracy: 0.6323
Epoch 88/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6104 -
accuracy: 0.6504 - val_loss: 0.6343 - val_accuracy: 0.6296
Epoch 89/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6101 -
accuracy: 0.6504 - val_loss: 0.6352 - val_accuracy: 0.6304
Epoch 90/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6097 -
accuracy: 0.6495 - val_loss: 0.6382 - val_accuracy: 0.6310
Epoch 91/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6096 -
accuracy: 0.6508 - val_loss: 0.6369 - val_accuracy: 0.6302
Epoch 92/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6094 -
accuracy: 0.6485 - val_loss: 0.6352 - val_accuracy: 0.6335
Epoch 93/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6093 -
accuracy: 0.6492 - val_loss: 0.6352 - val_accuracy: 0.6302
Epoch 94/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6103 -
```

```
accuracy: 0.6516 - val_loss: 0.6358 - val_accuracy: 0.6319
Epoch 95/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6097 -
accuracy: 0.6518 - val_loss: 0.6398 - val_accuracy: 0.6358
Epoch 96/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6091 -
accuracy: 0.6512 - val_loss: 0.6363 - val_accuracy: 0.6306
Epoch 97/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6088 -
accuracy: 0.6495 - val_loss: 0.6362 - val_accuracy: 0.6354
Epoch 98/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6096 -
accuracy: 0.6514 - val_loss: 0.6352 - val_accuracy: 0.6319
Epoch 99/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6082 -
accuracy: 0.6519 - val_loss: 0.6355 - val_accuracy: 0.6310
Epoch 100/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6080 -
accuracy: 0.6526 - val_loss: 0.6365 - val_accuracy: 0.6343
Epoch 101/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6077 -
accuracy: 0.6508 - val_loss: 0.6365 - val_accuracy: 0.6323
Epoch 102/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6085 -
accuracy: 0.6539 - val_loss: 0.6371 - val_accuracy: 0.6306
Epoch 103/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6084 -
accuracy: 0.6525 - val_loss: 0.6358 - val_accuracy: 0.6354
Epoch 104/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6081 -
accuracy: 0.6547 - val_loss: 0.6371 - val_accuracy: 0.6335
Epoch 105/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6078 -
accuracy: 0.6513 - val_loss: 0.6369 - val_accuracy: 0.6333
Epoch 106/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6070 -
accuracy: 0.6535 - val_loss: 0.6353 - val_accuracy: 0.6298
Epoch 107/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6074 -
accuracy: 0.6519 - val_loss: 0.6354 - val_accuracy: 0.6354
Epoch 108/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6069 -
accuracy: 0.6548 - val_loss: 0.6354 - val_accuracy: 0.6337
Epoch 109/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6064 -
accuracy: 0.6551 - val_loss: 0.6356 - val_accuracy: 0.6323
Epoch 110/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6061 -
accuracy: 0.6549 - val_loss: 0.6373 - val_accuracy: 0.6339
Epoch 111/200
```

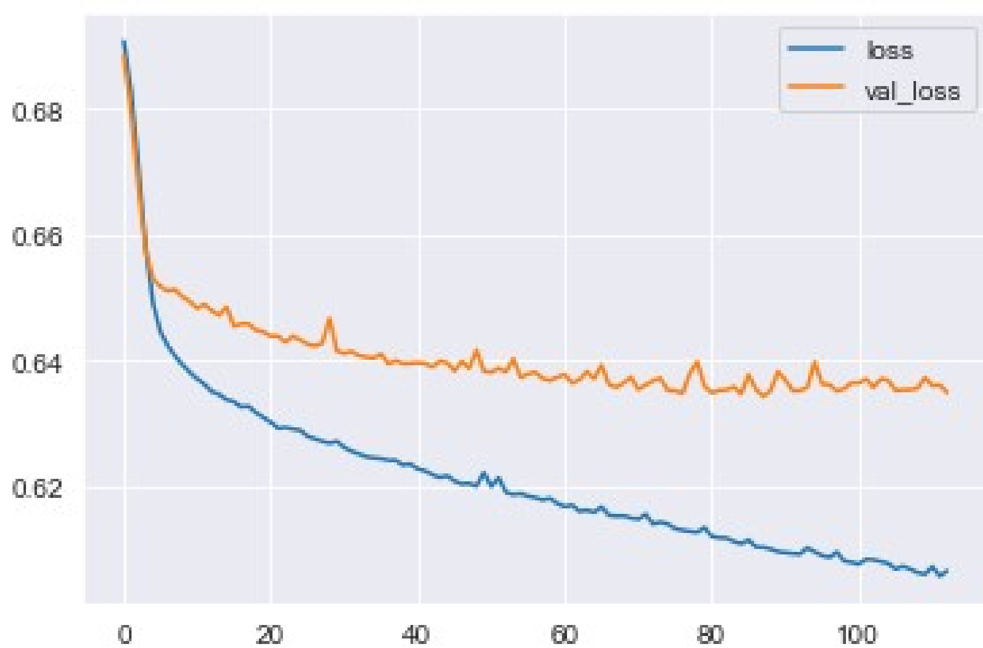
```
44/44 [==============================] - 0s 1ms/step - loss: 0.6074 -
accuracy: 0.6545 - val_loss: 0.6360 - val_accuracy: 0.6348
Epoch 112/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6059 -
accuracy: 0.6577 - val_loss: 0.6361 - val_accuracy: 0.6329
Epoch 113/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6067 -
accuracy: 0.6552 - val_loss: 0.6349 - val_accuracy: 0.6366
Epoch 00113: early stopping

<keras.callbacks.History at 0x2a405216860>
```

Model Evaluation and Validation

```
pd.DataFrame(model.history.history)[['loss','val_loss']].plot()

<matplotlib.axes._subplots.AxesSubplot at 0x2a4033789b0>
```



This validation result, the Loss plot, shows us the model is overfitted.

```
predictions = (model.predict(X_test) > 0.5).astype("int32")

print(
        confusion_matrix(y_test,predictions),
        '\n',
        classification_report(y_test,predictions)
)

[[1655  782]
 [ 972 1418]]
               precision    recall  f1-score   support
```

```
           0       0.63      0.68      0.65      2437
           1       0.64      0.59      0.62      2390

    accuracy                           0.64      4827
   macro avg       0.64      0.64      0.64      4827
weighted avg       0.64      0.64      0.64      4827
```

The model's overall f1-score for accuracy is 0.65. Still, there are type 2 errors (972) in the prediction.

Model Refinement

Two ways of refining the model we will try here. Add Dropout layers to bring down the overfitting OR Lower the cut-off line in binary prediction to reduce the Type 2 error, at the cost of increasing Type 1 error. In the LendingClub case, Type 2 error is the more serious problem because it devastates its balance sheet, while Type 1 error is not a very big deal.

```
model_new = Sequential()

model_new.add(
        Dense(19, activation='relu')
)

model_new.add(Dropout(0.2))

model_new.add(
        Dense(10, activation='relu')
)

model_new.add(Dropout(0.2))

model_new.add(
        Dense(5, activation='relu')
)

model_new.add(Dropout(0.2))

model_new.add(
        Dense(1, activation='sigmoid')
)

model_new.compile(
        optimizer='adam',
        loss='binary_crossentropy',
        metrics=['binary_accuracy']
)
```

```python
model_new.fit(
        X_train,
        y_train,
        epochs=200,
        batch_size=256,
        validation_data=(X_test, y_test),
         callbacks=[early_stop]
)
```

```
Epoch 1/200
44/44 [==============================] - 1s 4ms/step - loss: 0.6912 -
binary_accuracy: 0.5380 - val_loss: 0.6896 - val_binary_accuracy:
0.5780
Epoch 2/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6876 -
binary_accuracy: 0.5537 - val_loss: 0.6832 - val_binary_accuracy:
0.5927
Epoch 3/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6809 -
binary_accuracy: 0.5649 - val_loss: 0.6749 - val_binary_accuracy:
0.6070
Epoch 4/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6735 -
binary_accuracy: 0.5788 - val_loss: 0.6666 - val_binary_accuracy:
0.6091
Epoch 5/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6681 -
binary_accuracy: 0.5957 - val_loss: 0.6624 - val_binary_accuracy:
0.6070
Epoch 6/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6659 -
binary_accuracy: 0.5934 - val_loss: 0.6605 - val_binary_accuracy:
0.6118
Epoch 7/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6625 -
binary_accuracy: 0.6029 - val_loss: 0.6589 - val_binary_accuracy:
0.6095
Epoch 8/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6617 -
binary_accuracy: 0.6087 - val_loss: 0.6586 - val_binary_accuracy:
0.6145
Epoch 9/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6615 -
binary_accuracy: 0.6093 - val_loss: 0.6585 - val_binary_accuracy:
0.6078
Epoch 10/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6601 -
binary_accuracy: 0.6062 - val_loss: 0.6573 - val_binary_accuracy:
```

```
0.6053
Epoch 11/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6568 -
binary_accuracy: 0.6156 - val_loss: 0.6564 - val_binary_accuracy:
0.6087
Epoch 12/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6579 -
binary_accuracy: 0.6133 - val_loss: 0.6561 - val_binary_accuracy:
0.6107
Epoch 13/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6576 -
binary_accuracy: 0.6206 - val_loss: 0.6556 - val_binary_accuracy:
0.6095
Epoch 14/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6550 -
binary_accuracy: 0.6175 - val_loss: 0.6555 - val_binary_accuracy:
0.6116
Epoch 15/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6567 -
binary_accuracy: 0.6148 - val_loss: 0.6551 - val_binary_accuracy:
0.6138
Epoch 16/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6540 -
binary_accuracy: 0.6172 - val_loss: 0.6541 - val_binary_accuracy:
0.6153
Epoch 17/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6533 -
binary_accuracy: 0.6181 - val_loss: 0.6535 - val_binary_accuracy:
0.6165
Epoch 18/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6506 -
binary_accuracy: 0.6204 - val_loss: 0.6530 - val_binary_accuracy:
0.6153
Epoch 19/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6533 -
binary_accuracy: 0.6203 - val_loss: 0.6530 - val_binary_accuracy:
0.6178
Epoch 20/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6518 -
binary_accuracy: 0.6189 - val_loss: 0.6523 - val_binary_accuracy:
0.6165
Epoch 21/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6507 -
binary_accuracy: 0.6188 - val_loss: 0.6514 - val_binary_accuracy:
0.6165
Epoch 22/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6483 -
binary_accuracy: 0.6182 - val_loss: 0.6511 - val_binary_accuracy:
0.6116
Epoch 23/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6489 -
binary_accuracy: 0.6200 - val_loss: 0.6510 - val_binary_accuracy:
0.6163
Epoch 24/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6501 -
binary_accuracy: 0.6198 - val_loss: 0.6505 - val_binary_accuracy:
0.6169
Epoch 25/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6467 -
binary_accuracy: 0.6169 - val_loss: 0.6501 - val_binary_accuracy:
0.6174
Epoch 26/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6476 -
binary_accuracy: 0.6229 - val_loss: 0.6495 - val_binary_accuracy:
0.6172
Epoch 27/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6483 -
binary_accuracy: 0.6184 - val_loss: 0.6500 - val_binary_accuracy:
0.6169
Epoch 28/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6482 -
binary_accuracy: 0.6176 - val_loss: 0.6503 - val_binary_accuracy:
0.6138
Epoch 29/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6477 -
binary_accuracy: 0.6247 - val_loss: 0.6495 - val_binary_accuracy:
0.6134
Epoch 30/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6486 -
binary_accuracy: 0.6239 - val_loss: 0.6494 - val_binary_accuracy:
0.6213
Epoch 31/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6484 -
binary_accuracy: 0.6197 - val_loss: 0.6495 - val_binary_accuracy:
0.6225
Epoch 32/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6447 -
binary_accuracy: 0.6266 - val_loss: 0.6487 - val_binary_accuracy:
0.6174
Epoch 33/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6460 -
binary_accuracy: 0.6212 - val_loss: 0.6486 - val_binary_accuracy:
0.6230
Epoch 34/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6460 -
binary_accuracy: 0.6243 - val_loss: 0.6490 - val_binary_accuracy:
0.6215
Epoch 35/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6462 -
binary_accuracy: 0.6213 - val_loss: 0.6489 - val_binary_accuracy:
```

```
0.6151
Epoch 36/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6457 -
binary_accuracy: 0.6217 - val_loss: 0.6483 - val_binary_accuracy:
0.6236
Epoch 37/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6458 -
binary_accuracy: 0.6303 - val_loss: 0.6485 - val_binary_accuracy:
0.6230
Epoch 38/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6433 -
binary_accuracy: 0.6259 - val_loss: 0.6479 - val_binary_accuracy:
0.6207
Epoch 39/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6442 -
binary_accuracy: 0.6276 - val_loss: 0.6475 - val_binary_accuracy:
0.6240
Epoch 40/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6443 -
binary_accuracy: 0.6295 - val_loss: 0.6477 - val_binary_accuracy:
0.6219
Epoch 41/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6438 -
binary_accuracy: 0.6266 - val_loss: 0.6473 - val_binary_accuracy:
0.6244
Epoch 42/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6442 -
binary_accuracy: 0.6231 - val_loss: 0.6470 - val_binary_accuracy:
0.6275
Epoch 43/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6439 -
binary_accuracy: 0.6272 - val_loss: 0.6470 - val_binary_accuracy:
0.6261
Epoch 44/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6438 -
binary_accuracy: 0.6290 - val_loss: 0.6469 - val_binary_accuracy:
0.6246
Epoch 45/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6446 -
binary_accuracy: 0.6291 - val_loss: 0.6472 - val_binary_accuracy:
0.6248
Epoch 46/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6432 -
binary_accuracy: 0.6267 - val_loss: 0.6470 - val_binary_accuracy:
0.6203
Epoch 47/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6413 -
binary_accuracy: 0.6279 - val_loss: 0.6466 - val_binary_accuracy:
0.6223
Epoch 48/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6415 -
binary_accuracy: 0.6332 - val_loss: 0.6472 - val_binary_accuracy:
0.6167
Epoch 49/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6424 -
binary_accuracy: 0.6295 - val_loss: 0.6456 - val_binary_accuracy:
0.6265
Epoch 50/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6398 -
binary_accuracy: 0.6309 - val_loss: 0.6458 - val_binary_accuracy:
0.6248
Epoch 51/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6403 -
binary_accuracy: 0.6283 - val_loss: 0.6461 - val_binary_accuracy:
0.6178
Epoch 52/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6419 -
binary_accuracy: 0.6301 - val_loss: 0.6455 - val_binary_accuracy:
0.6240
Epoch 53/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6420 -
binary_accuracy: 0.6311 - val_loss: 0.6449 - val_binary_accuracy:
0.6236
Epoch 54/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6398 -
binary_accuracy: 0.6339 - val_loss: 0.6444 - val_binary_accuracy:
0.6213
Epoch 55/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6388 -
binary_accuracy: 0.6279 - val_loss: 0.6444 - val_binary_accuracy:
0.6254
Epoch 56/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6392 -
binary_accuracy: 0.6343 - val_loss: 0.6441 - val_binary_accuracy:
0.6242
Epoch 57/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6377 -
binary_accuracy: 0.6335 - val_loss: 0.6431 - val_binary_accuracy:
0.6285
Epoch 58/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6391 -
binary_accuracy: 0.6321 - val_loss: 0.6430 - val_binary_accuracy:
0.6277
Epoch 59/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6385 -
binary_accuracy: 0.6258 - val_loss: 0.6433 - val_binary_accuracy:
0.6236
Epoch 60/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6387 -
binary_accuracy: 0.6330 - val_loss: 0.6427 - val_binary_accuracy:
```

```
0.6271
Epoch 61/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6363 -
binary_accuracy: 0.6323 - val_loss: 0.6426 - val_binary_accuracy:
0.6271
Epoch 62/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6371 -
binary_accuracy: 0.6340 - val_loss: 0.6422 - val_binary_accuracy:
0.6244
Epoch 63/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6364 -
binary_accuracy: 0.6307 - val_loss: 0.6418 - val_binary_accuracy:
0.6238
Epoch 64/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6366 -
binary_accuracy: 0.6329 - val_loss: 0.6420 - val_binary_accuracy:
0.6290
Epoch 65/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6341 -
binary_accuracy: 0.6300 - val_loss: 0.6415 - val_binary_accuracy:
0.6242
Epoch 66/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6371 -
binary_accuracy: 0.6343 - val_loss: 0.6416 - val_binary_accuracy:
0.6252
Epoch 67/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6348 -
binary_accuracy: 0.6352 - val_loss: 0.6411 - val_binary_accuracy:
0.6252
Epoch 68/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6336 -
binary_accuracy: 0.6326 - val_loss: 0.6415 - val_binary_accuracy:
0.6269
Epoch 69/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6345 -
binary_accuracy: 0.6348 - val_loss: 0.6409 - val_binary_accuracy:
0.6232
Epoch 70/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6335 -
binary_accuracy: 0.6312 - val_loss: 0.6411 - val_binary_accuracy:
0.6254
Epoch 71/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6331 -
binary_accuracy: 0.6358 - val_loss: 0.6411 - val_binary_accuracy:
0.6288
Epoch 72/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6329 -
binary_accuracy: 0.6367 - val_loss: 0.6405 - val_binary_accuracy:
0.6252
Epoch 73/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6346 -
binary_accuracy: 0.6328 - val_loss: 0.6411 - val_binary_accuracy:
0.6256
Epoch 74/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6333 -
binary_accuracy: 0.6365 - val_loss: 0.6406 - val_binary_accuracy:
0.6273
Epoch 75/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6322 -
binary_accuracy: 0.6364 - val_loss: 0.6408 - val_binary_accuracy:
0.6275
Epoch 76/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6357 -
binary_accuracy: 0.6324 - val_loss: 0.6404 - val_binary_accuracy:
0.6279
Epoch 77/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6325 -
binary_accuracy: 0.6359 - val_loss: 0.6402 - val_binary_accuracy:
0.6271
Epoch 78/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6333 -
binary_accuracy: 0.6308 - val_loss: 0.6393 - val_binary_accuracy:
0.6254
Epoch 79/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6323 -
binary_accuracy: 0.6348 - val_loss: 0.6398 - val_binary_accuracy:
0.6261
Epoch 80/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6333 -
binary_accuracy: 0.6369 - val_loss: 0.6394 - val_binary_accuracy:
0.6290
Epoch 81/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6331 -
binary_accuracy: 0.6356 - val_loss: 0.6389 - val_binary_accuracy:
0.6275
Epoch 82/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6339 -
binary_accuracy: 0.6330 - val_loss: 0.6391 - val_binary_accuracy:
0.6265
Epoch 83/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6326 -
binary_accuracy: 0.6393 - val_loss: 0.6388 - val_binary_accuracy:
0.6281
Epoch 84/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6323 -
binary_accuracy: 0.6356 - val_loss: 0.6391 - val_binary_accuracy:
0.6273
Epoch 85/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6332 -
binary_accuracy: 0.6394 - val_loss: 0.6402 - val_binary_accuracy:
```

```
0.6292
Epoch 86/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6298 -
binary_accuracy: 0.6368 - val_loss: 0.6392 - val_binary_accuracy:
0.6300
Epoch 87/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6307 -
binary_accuracy: 0.6358 - val_loss: 0.6388 - val_binary_accuracy:
0.6298
Epoch 88/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6326 -
binary_accuracy: 0.6413 - val_loss: 0.6385 - val_binary_accuracy:
0.6304
Epoch 89/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6346 -
binary_accuracy: 0.6349 - val_loss: 0.6382 - val_binary_accuracy:
0.6294
Epoch 90/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6303 -
binary_accuracy: 0.6354 - val_loss: 0.6384 - val_binary_accuracy:
0.6263
Epoch 91/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6312 -
binary_accuracy: 0.6370 - val_loss: 0.6384 - val_binary_accuracy:
0.6269
Epoch 92/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6301 -
binary_accuracy: 0.6375 - val_loss: 0.6377 - val_binary_accuracy:
0.6331
Epoch 93/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6298 -
binary_accuracy: 0.6394 - val_loss: 0.6383 - val_binary_accuracy:
0.6275
Epoch 94/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6313 -
binary_accuracy: 0.6408 - val_loss: 0.6386 - val_binary_accuracy:
0.6261
Epoch 95/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6302 -
binary_accuracy: 0.6368 - val_loss: 0.6371 - val_binary_accuracy:
0.6263
Epoch 96/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6292 -
binary_accuracy: 0.6375 - val_loss: 0.6375 - val_binary_accuracy:
0.6271
Epoch 97/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6303 -
binary_accuracy: 0.6410 - val_loss: 0.6374 - val_binary_accuracy:
0.6329
Epoch 98/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6290 -
binary_accuracy: 0.6369 - val_loss: 0.6376 - val_binary_accuracy:
0.6248
Epoch 99/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6314 -
binary_accuracy: 0.6387 - val_loss: 0.6367 - val_binary_accuracy:
0.6323
Epoch 100/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6298 -
binary_accuracy: 0.6356 - val_loss: 0.6374 - val_binary_accuracy:
0.6290
Epoch 101/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6297 -
binary_accuracy: 0.6418 - val_loss: 0.6367 - val_binary_accuracy:
0.6310
Epoch 102/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6287 -
binary_accuracy: 0.6399 - val_loss: 0.6366 - val_binary_accuracy:
0.6292
Epoch 103/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6284 -
binary_accuracy: 0.6354 - val_loss: 0.6360 - val_binary_accuracy:
0.6312
Epoch 104/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6297 -
binary_accuracy: 0.6345 - val_loss: 0.6370 - val_binary_accuracy:
0.6275
Epoch 105/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6309 -
binary_accuracy: 0.6421 - val_loss: 0.6365 - val_binary_accuracy:
0.6288
Epoch 106/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6295 -
binary_accuracy: 0.6394 - val_loss: 0.6356 - val_binary_accuracy:
0.6304
Epoch 107/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6276 -
binary_accuracy: 0.6422 - val_loss: 0.6360 - val_binary_accuracy:
0.6306
Epoch 108/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6256 -
binary_accuracy: 0.6433 - val_loss: 0.6360 - val_binary_accuracy:
0.6275
Epoch 109/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6283 -
binary_accuracy: 0.6389 - val_loss: 0.6360 - val_binary_accuracy:
0.6273
Epoch 110/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6290 -
binary_accuracy: 0.6387 - val_loss: 0.6358 - val_binary_accuracy:
```

```
0.6304
Epoch 111/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6269 -
binary_accuracy: 0.6379 - val_loss: 0.6359 - val_binary_accuracy:
0.6296
Epoch 112/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6290 -
binary_accuracy: 0.6346 - val_loss: 0.6354 - val_binary_accuracy:
0.6343
Epoch 113/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6263 -
binary_accuracy: 0.6443 - val_loss: 0.6354 - val_binary_accuracy:
0.6343
Epoch 114/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6280 -
binary_accuracy: 0.6389 - val_loss: 0.6348 - val_binary_accuracy:
0.6331
Epoch 115/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6273 -
binary_accuracy: 0.6457 - val_loss: 0.6351 - val_binary_accuracy:
0.6346
Epoch 116/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6237 -
binary_accuracy: 0.6466 - val_loss: 0.6352 - val_binary_accuracy:
0.6333
Epoch 117/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6266 -
binary_accuracy: 0.6374 - val_loss: 0.6352 - val_binary_accuracy:
0.6323
Epoch 118/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6281 -
binary_accuracy: 0.6406 - val_loss: 0.6345 - val_binary_accuracy:
0.6321
Epoch 119/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6254 -
binary_accuracy: 0.6448 - val_loss: 0.6345 - val_binary_accuracy:
0.6375
Epoch 120/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6265 -
binary_accuracy: 0.6371 - val_loss: 0.6345 - val_binary_accuracy:
0.6352
Epoch 121/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6256 -
binary_accuracy: 0.6377 - val_loss: 0.6349 - val_binary_accuracy:
0.6372
Epoch 122/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6243 -
binary_accuracy: 0.6413 - val_loss: 0.6336 - val_binary_accuracy:
0.6350
Epoch 123/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6252 -
binary_accuracy: 0.6378 - val_loss: 0.6339 - val_binary_accuracy:
0.6379
Epoch 124/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6251 -
binary_accuracy: 0.6432 - val_loss: 0.6338 - val_binary_accuracy:
0.6406
Epoch 125/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6281 -
binary_accuracy: 0.6435 - val_loss: 0.6337 - val_binary_accuracy:
0.6381
Epoch 126/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6280 -
binary_accuracy: 0.6412 - val_loss: 0.6334 - val_binary_accuracy:
0.6308
Epoch 127/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6252 -
binary_accuracy: 0.6419 - val_loss: 0.6333 - val_binary_accuracy:
0.6317
Epoch 128/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6277 -
binary_accuracy: 0.6471 - val_loss: 0.6331 - val_binary_accuracy:
0.6412
Epoch 129/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6277 -
binary_accuracy: 0.6380 - val_loss: 0.6333 - val_binary_accuracy:
0.6368
Epoch 130/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6248 -
binary_accuracy: 0.6446 - val_loss: 0.6332 - val_binary_accuracy:
0.6360
Epoch 131/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6225 -
binary_accuracy: 0.6422 - val_loss: 0.6327 - val_binary_accuracy:
0.6383
Epoch 132/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6262 -
binary_accuracy: 0.6457 - val_loss: 0.6333 - val_binary_accuracy:
0.6333
Epoch 133/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6257 -
binary_accuracy: 0.6423 - val_loss: 0.6326 - val_binary_accuracy:
0.6366
Epoch 134/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6256 -
binary_accuracy: 0.6436 - val_loss: 0.6324 - val_binary_accuracy:
0.6348
Epoch 135/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6252 -
binary_accuracy: 0.6473 - val_loss: 0.6327 - val_binary_accuracy:
```

```
0.6300
Epoch 136/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6275 -
binary_accuracy: 0.6402 - val_loss: 0.6313 - val_binary_accuracy:
0.6379
Epoch 137/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6249 -
binary_accuracy: 0.6439 - val_loss: 0.6324 - val_binary_accuracy:
0.6356
Epoch 138/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6236 -
binary_accuracy: 0.6457 - val_loss: 0.6314 - val_binary_accuracy:
0.6350
Epoch 139/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6234 -
binary_accuracy: 0.6477 - val_loss: 0.6317 - val_binary_accuracy:
0.6350
Epoch 140/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6208 -
binary_accuracy: 0.6447 - val_loss: 0.6315 - val_binary_accuracy:
0.6356
Epoch 141/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6229 -
binary_accuracy: 0.6504 - val_loss: 0.6309 - val_binary_accuracy:
0.6397
Epoch 142/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6228 -
binary_accuracy: 0.6418 - val_loss: 0.6313 - val_binary_accuracy:
0.6381
Epoch 143/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6254 -
binary_accuracy: 0.6453 - val_loss: 0.6313 - val_binary_accuracy:
0.6352
Epoch 144/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6219 -
binary_accuracy: 0.6511 - val_loss: 0.6313 - val_binary_accuracy:
0.6352
Epoch 145/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6270 -
binary_accuracy: 0.6427 - val_loss: 0.6307 - val_binary_accuracy:
0.6397
Epoch 146/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6237 -
binary_accuracy: 0.6535 - val_loss: 0.6315 - val_binary_accuracy:
0.6366
Epoch 147/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6240 -
binary_accuracy: 0.6475 - val_loss: 0.6300 - val_binary_accuracy:
0.6399
Epoch 148/200
```

```
44/44 [==============================] - 0s 3ms/step - loss: 0.6226 -
binary_accuracy: 0.6484 - val_loss: 0.6312 - val_binary_accuracy:
0.6379
Epoch 149/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6219 -
binary_accuracy: 0.6489 - val_loss: 0.6311 - val_binary_accuracy:
0.6428
Epoch 150/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6237 -
binary_accuracy: 0.6468 - val_loss: 0.6313 - val_binary_accuracy:
0.6383
Epoch 151/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6199 -
binary_accuracy: 0.6478 - val_loss: 0.6302 - val_binary_accuracy:
0.6397
Epoch 152/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6223 -
binary_accuracy: 0.6487 - val_loss: 0.6305 - val_binary_accuracy:
0.6401
Epoch 153/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6249 -
binary_accuracy: 0.6420 - val_loss: 0.6307 - val_binary_accuracy:
0.6391
Epoch 154/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6234 -
binary_accuracy: 0.6470 - val_loss: 0.6302 - val_binary_accuracy:
0.6426
Epoch 155/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6220 -
binary_accuracy: 0.6453 - val_loss: 0.6300 - val_binary_accuracy:
0.6393
Epoch 156/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6207 -
binary_accuracy: 0.6496 - val_loss: 0.6317 - val_binary_accuracy:
0.6387
Epoch 157/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6180 -
binary_accuracy: 0.6494 - val_loss: 0.6307 - val_binary_accuracy:
0.6435
Epoch 158/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6222 -
binary_accuracy: 0.6446 - val_loss: 0.6312 - val_binary_accuracy:
0.6391
Epoch 159/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6205 -
binary_accuracy: 0.6452 - val_loss: 0.6323 - val_binary_accuracy:
0.6354
Epoch 160/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6256 -
binary_accuracy: 0.6400 - val_loss: 0.6305 - val_binary_accuracy:
```

```
0.6389
Epoch 161/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6208 -
binary_accuracy: 0.6505 - val_loss: 0.6313 - val_binary_accuracy:
0.6383
Epoch 162/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6225 -
binary_accuracy: 0.6410 - val_loss: 0.6298 - val_binary_accuracy:
0.6368
Epoch 163/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6226 -
binary_accuracy: 0.6472 - val_loss: 0.6295 - val_binary_accuracy:
0.6387
Epoch 164/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6213 -
binary_accuracy: 0.6494 - val_loss: 0.6302 - val_binary_accuracy:
0.6414
Epoch 165/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6218 -
binary_accuracy: 0.6448 - val_loss: 0.6295 - val_binary_accuracy:
0.6383
Epoch 166/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6177 -
binary_accuracy: 0.6536 - val_loss: 0.6292 - val_binary_accuracy:
0.6375
Epoch 167/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6214 -
binary_accuracy: 0.6522 - val_loss: 0.6290 - val_binary_accuracy:
0.6451
Epoch 168/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6198 -
binary_accuracy: 0.6469 - val_loss: 0.6297 - val_binary_accuracy:
0.6399
Epoch 169/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6167 -
binary_accuracy: 0.6559 - val_loss: 0.6293 - val_binary_accuracy:
0.6343
Epoch 170/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6193 -
binary_accuracy: 0.6486 - val_loss: 0.6290 - val_binary_accuracy:
0.6387
Epoch 171/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6210 -
binary_accuracy: 0.6527 - val_loss: 0.6297 - val_binary_accuracy:
0.6395
Epoch 172/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6209 -
binary_accuracy: 0.6489 - val_loss: 0.6291 - val_binary_accuracy:
0.6370
Epoch 173/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6199 -
binary_accuracy: 0.6496 - val_loss: 0.6284 - val_binary_accuracy:
0.6389
Epoch 174/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6220 -
binary_accuracy: 0.6448 - val_loss: 0.6283 - val_binary_accuracy:
0.6412
Epoch 175/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6191 -
binary_accuracy: 0.6510 - val_loss: 0.6287 - val_binary_accuracy:
0.6410
Epoch 176/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6206 -
binary_accuracy: 0.6490 - val_loss: 0.6285 - val_binary_accuracy:
0.6393
Epoch 177/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6214 -
binary_accuracy: 0.6467 - val_loss: 0.6292 - val_binary_accuracy:
0.6404
Epoch 178/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6178 -
binary_accuracy: 0.6500 - val_loss: 0.6284 - val_binary_accuracy:
0.6408
Epoch 179/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6169 -
binary_accuracy: 0.6515 - val_loss: 0.6288 - val_binary_accuracy:
0.6414
Epoch 180/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6225 -
binary_accuracy: 0.6484 - val_loss: 0.6290 - val_binary_accuracy:
0.6418
Epoch 181/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6205 -
binary_accuracy: 0.6528 - val_loss: 0.6286 - val_binary_accuracy:
0.6383
Epoch 182/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6197 -
binary_accuracy: 0.6475 - val_loss: 0.6280 - val_binary_accuracy:
0.6420
Epoch 183/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6196 -
binary_accuracy: 0.6497 - val_loss: 0.6279 - val_binary_accuracy:
0.6387
Epoch 184/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6199 -
binary_accuracy: 0.6540 - val_loss: 0.6285 - val_binary_accuracy:
0.6399
Epoch 185/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6220 -
binary_accuracy: 0.6480 - val_loss: 0.6292 - val_binary_accuracy:
```

```
0.6383
Epoch 186/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6176 -
binary_accuracy: 0.6528 - val_loss: 0.6286 - val_binary_accuracy:
0.6383
Epoch 187/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6189 -
binary_accuracy: 0.6484 - val_loss: 0.6274 - val_binary_accuracy:
0.6443
Epoch 188/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6177 -
binary_accuracy: 0.6551 - val_loss: 0.6273 - val_binary_accuracy:
0.6428
Epoch 189/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6179 -
binary_accuracy: 0.6530 - val_loss: 0.6274 - val_binary_accuracy:
0.6401
Epoch 190/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6153 -
binary_accuracy: 0.6581 - val_loss: 0.6282 - val_binary_accuracy:
0.6358
Epoch 191/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6172 -
binary_accuracy: 0.6551 - val_loss: 0.6292 - val_binary_accuracy:
0.6360
Epoch 192/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6180 -
binary_accuracy: 0.6527 - val_loss: 0.6279 - val_binary_accuracy:
0.6422
Epoch 193/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6175 -
binary_accuracy: 0.6496 - val_loss: 0.6269 - val_binary_accuracy:
0.6430
Epoch 194/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6196 -
binary_accuracy: 0.6520 - val_loss: 0.6275 - val_binary_accuracy:
0.6408
Epoch 195/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6201 -
binary_accuracy: 0.6533 - val_loss: 0.6279 - val_binary_accuracy:
0.6383
Epoch 196/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6167 -
binary_accuracy: 0.6545 - val_loss: 0.6272 - val_binary_accuracy:
0.6389
Epoch 197/200
44/44 [==============================] - ETA: 0s - loss: 0.6192 -
binary_accuracy: 0.650 - 0s 2ms/step - loss: 0.6201 - binary_accuracy:
0.6481 - val_loss: 0.6286 - val_binary_accuracy: 0.6401
Epoch 198/200
```

```
44/44 [==============================] - 0s 2ms/step - loss: 0.6191 - binary_accuracy: 0.6552 - val_loss: 0.6287 - val_binary_accuracy: 0.6389
Epoch 199/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6174 - binary_accuracy: 0.6634 - val_loss: 0.6271 - val_binary_accuracy: 0.6449
Epoch 200/200
44/44 [==============================] - 0s 2ms/step - loss: 0.6176 - binary_accuracy: 0.6522 - val_loss: 0.6261 - val_binary_accuracy: 0.6439

<keras.callbacks.History at 0x2a40d53cb00>
```

```
pd.DataFrame(model_new.history.history)[['loss','val_loss']].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2a40d4f9e80>
```

The graph shows that, by adding in Dropout layers, we have reduced the overfitting issue compared with the old model.

```
# predictions_new = (model_new.predict_proba(X_test) >= 0.2).astype('int')
predictions_new = (model_new.predict(X_test) >= 0.2).astype('int')

# predictions_new=model.predict([testa,testb])

print(
        confusion_matrix(y_test,predictions_new),
        '\n',
```

```
        classification_report(y_test,predictions_new)
)

[[ 245 2192]
 [  39 2351]]
              precision    recall  f1-score   support

           0       0.86      0.10      0.18      2437
           1       0.52      0.98      0.68      2390

    accuracy                           0.54      4827
   macro avg       0.69      0.54      0.43      4827
weighted avg       0.69      0.54      0.43      4827
```

By changing the cut-off line to 0.2 (default is 0.5), we have dramatically brought down the Type 2 error.