# Lending_Club_Project_Solution

May 9, 2024

**Lending Club Loan Data Analysis  by Thato Tladi**

Course-end Project 3

**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.

**Tasks:** 1. Feature Transformation Transform categorical values into numerical values (discrete) 2. Exploratory data analysis of different factors of the dataset. 3. 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 4. 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

[1]:
```
# 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
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
```

[2]:
```
#loading dataset
df = pd.read_csv('loan_data.csv')
```

[3]:
```
#checking data head
df.head()
```

[3]:

| | credit.policy | purpose | int.rate | installment | log.annual.inc | dti | fico | days.with.cr.line | revol.bal | revol.util | inq.last.6mths | delinq.2yrs | pub.rec | not.fully.paid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | debt_consolidation | 0.1189 | 829.10 | 11.350407 | 19.48 | 737 | 5639.958333 | 28854 | 52.1 | 0 | 0 | 0 | 0 |
| 1 | 1 | credit_card | 0.1071 | 228.22 | 11.082143 | 14.29 | 707 | 2760.000000 | 33623 | 76.7 | 0 | 0 | 0 | 0 |
| 2 | 1 | debt_consolidation | 0.1357 | 366.86 | 10.373491 | 11.63 | 682 | 4710.000000 | 3511 | 25.6 | 1 | 0 | 0 | 0 |
| 3 | 1 | debt_consolidation | 0.1008 | 162.34 | 11.350407 | 8.10 | 712 | 2699.958333 | 33667 | 73.2 | 1 | 0 | 0 | 0 |
| 4 | 1 | credit_card | 0.1426 | 102.92 | 11.299732 | 14.97 | 667 | 4066.000000 | 4740 | 39.5 | 0 | 1 | 0 | 0 |

[4]:
```
#checking data info
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
```

[5]:
```
#Verying columns
df.columns
```

[5]:
```
Index(['credit.policy', 'purpose', 'int.rate', 'installment', 'log.annual.inc',
       'dti', 'fico', 'days.with.cr.line', 'revol.bal', 'revol.util',
       'inq.last.6mths', 'delinq.2yrs', 'pub.rec', 'not.fully.paid'],
      dtype='object')
```

[7]:
```
#checking data shape
df.shape
```

[7]:
```
(9578, 14)
```

[8]:
```
#statistical info
df.describe().transpose()
```

[8]:

| | 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
```

[9]:
```
#proportion of missing values
df['not.fully.paid'].isnull().mean()
```

[9]:
```
0.0
```

[10]:
```
#
df1=pd.get_dummies(df, columns=['purpose'])
```

[11]:
```
df1['log.annual.inc'] = np.exp(df1['log.annual.inc'])
```

[12]:
```
df1.head()
```

[12]:
```
   credit.policy  int.rate  installment  log.annual.inc    dti  fico  \
0              1    0.1189       829.10    85000.000385  19.48   737
1              1    0.1071       228.22    65000.000073  14.29   707
2              1    0.1357       366.86    31999.999943  11.63   682
3              1    0.1008       162.34    85000.000385   8.10   712
4              1    0.1426       102.92    80799.999636  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              False                False
1        0               0              False                 True
2        0               0              False                False
3        0               0              False                False
4        0               0              False                 True

   purpose_debt_consolidation  purpose_educational  purpose_home_improvement  \
0                        True                False                     False
1                       False                False                     False
2                        True                False                     False
3                        True                False                     False
4                       False                False                     False

   purpose_major_purchase  purpose_small_business
```

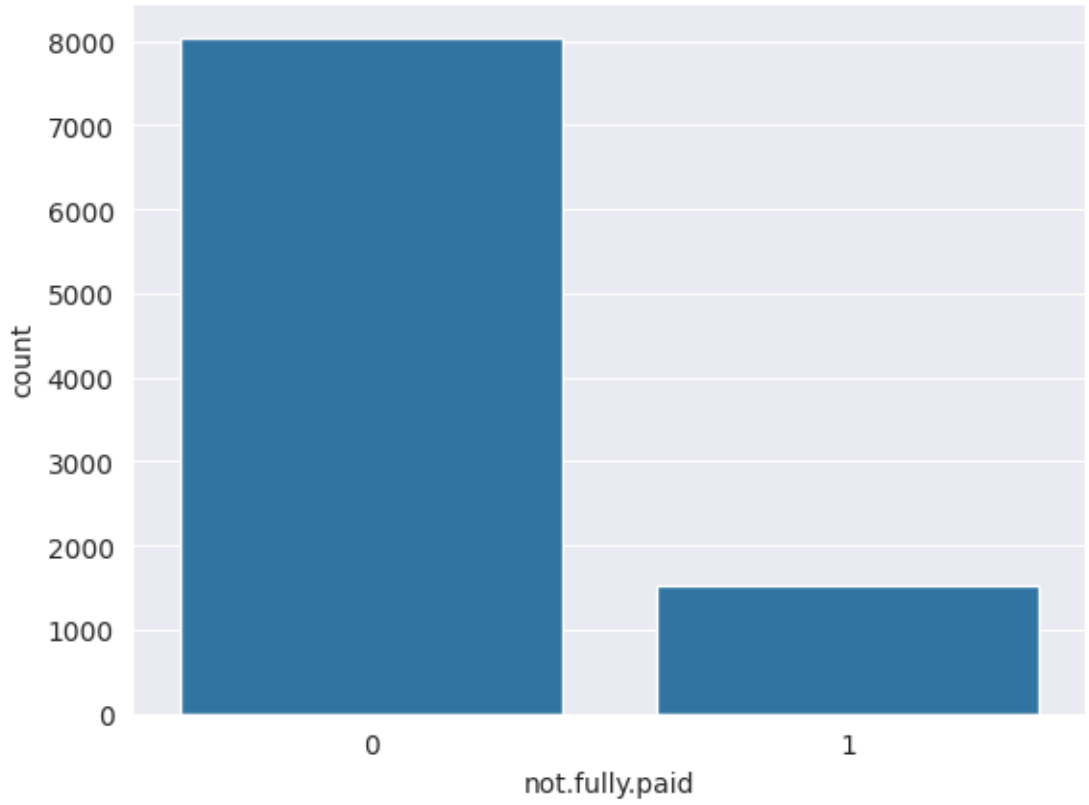
```
0                   False                   False
1                   False                   False
2                   False                   False
3                   False                   False
4                   False                   False
```

[13]:
```
df.groupby('not.fully.paid')['not.fully.paid'].count()/len(df)
```

[13]:
```
not.fully.paid
0    0.839946
1    0.160054
Name: not.fully.paid, dtype: float64
```

[14]:
```
sns.set_style('darkgrid')
sns.countplot(x='not.fully.paid', data=df)
```

[14]:
```
<Axes: xlabel='not.fully.paid', ylabel='count'>
```




The plot above shows this dataset is highly imbalanced.

*Using Oversampling approach to balance the dataset*

[16]:
```
count_class_0, count_class_1 = df['not.fully.paid'].value_counts()
```

[17]:
```
df_0 = df[df['not.fully.paid'] == 0]
df_1 = df[df['not.fully.paid'] == 1]
```

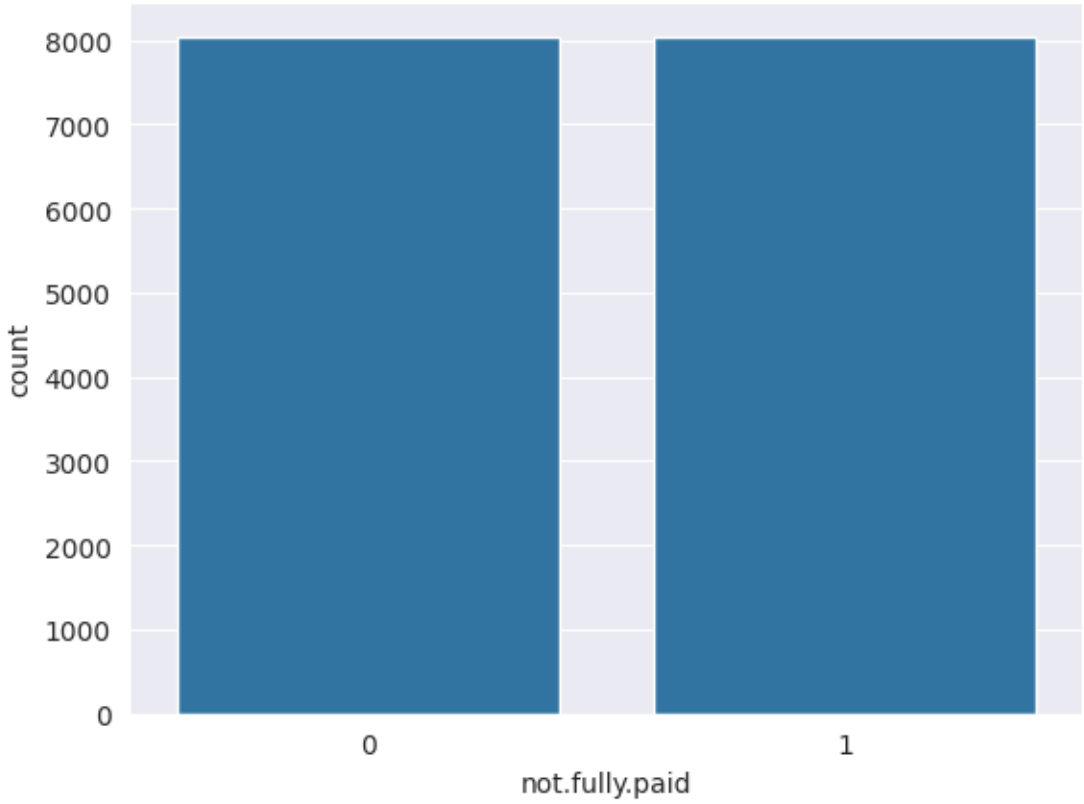
[18]:
```
df_1_over = df_1.sample(count_class_0, replace=True)
df_test_over = pd.concat([df_0, df_1_over], axis=0)
```

[19]:
```
print('Random over-sampling:')
print(df_test_over['not.fully.paid'].value_counts())
```

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

[20]:
```
#df_test_over['not.fully.paid'].value_counts().plot(kind='bar', title='Count␣
↪(not.fully.paid)')

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

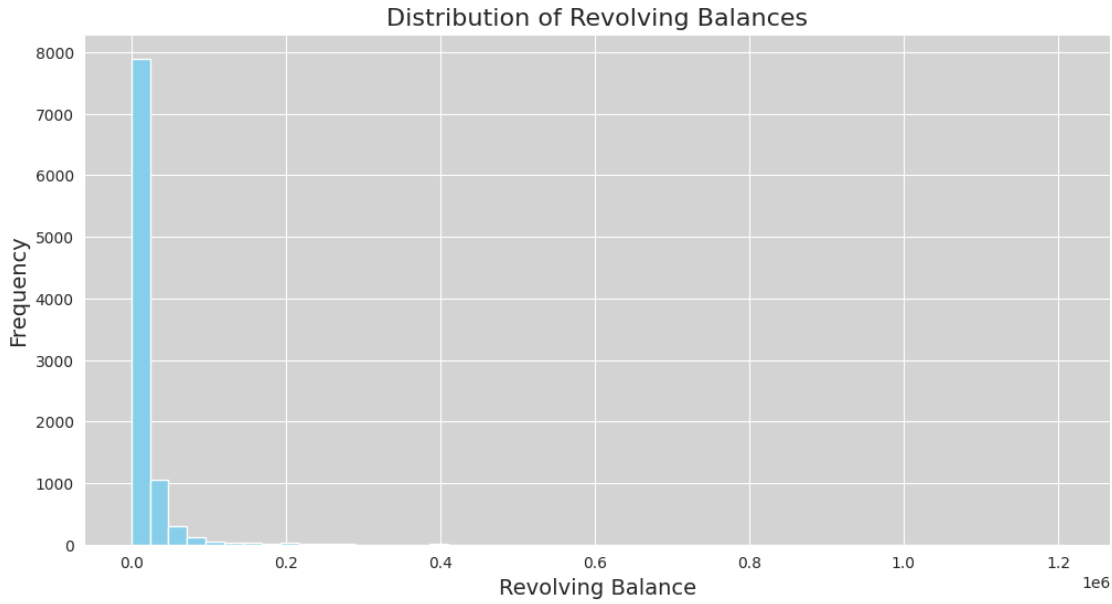
[20]: <Axes: xlabel='not.fully.paid', ylabel='count'>




Plot shows dataset is now balanced

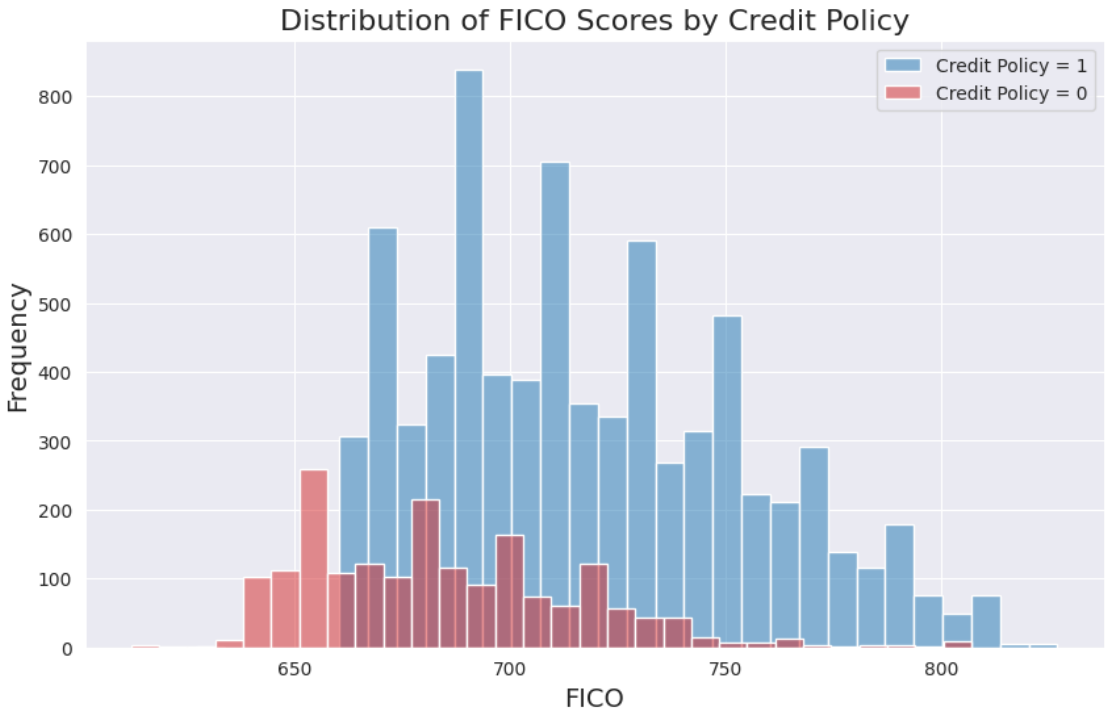
**Exploratory Data Analysis**

[23]:
```
plt.figure(figsize=[12, 6])
# Plot histogram with customizations
df['revol.bal'].hist(color='skyblue', bins=50)
# Add labels and title
plt.xlabel('Revolving Balance', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.title('Distribution of Revolving Balances', fontsize=16)
# Add gridlines
plt.grid(True)
# Set background color to grey
plt.gca().set_facecolor('lightgrey')
# Display the plot
plt.show()
```

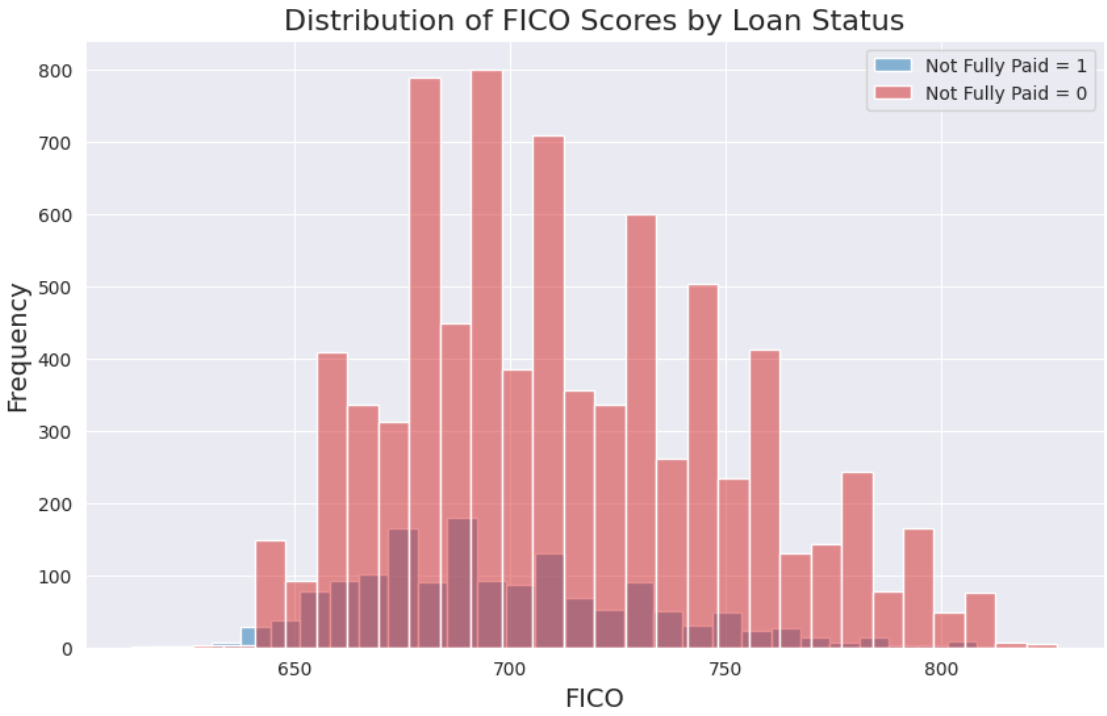



[24]:
```
df1=pd.get_dummies(df, columns=['purpose'])
```

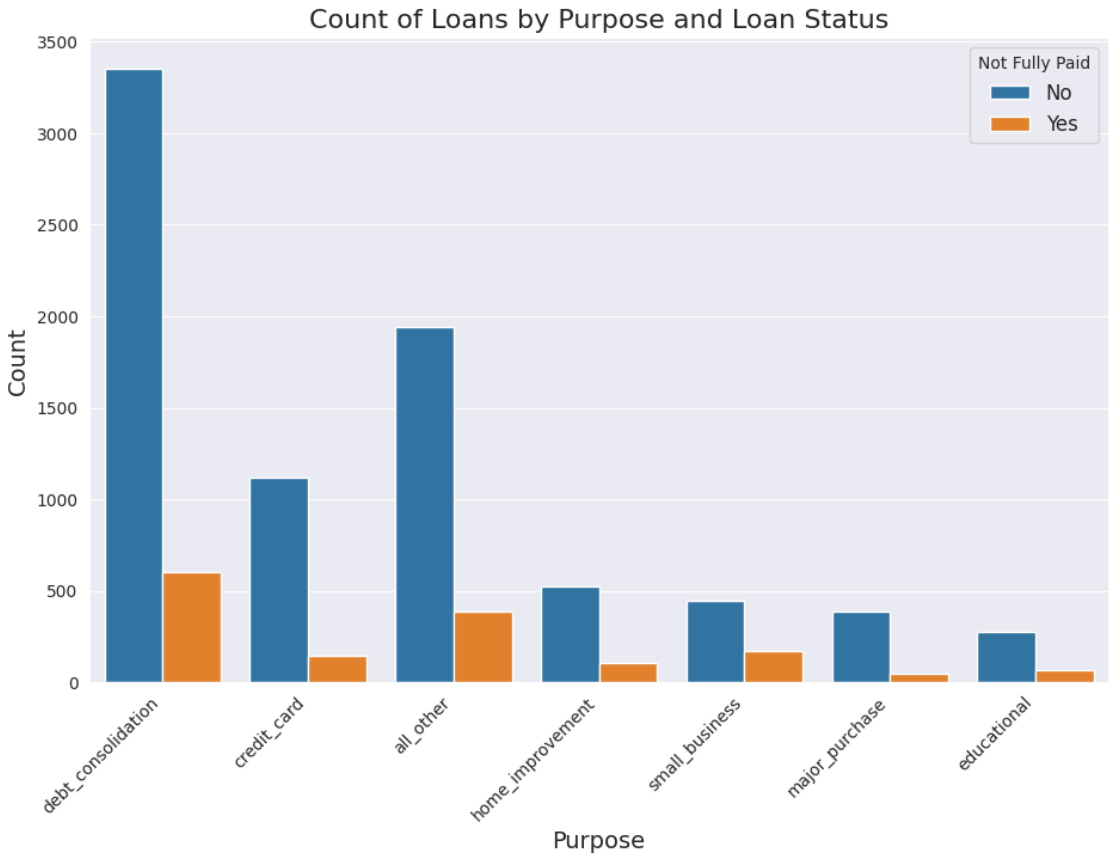
[26]:
```
plt.figure(figsize=(10, 6))
# Define custom color palette
colors = ["#1f77b4", "#d62728"]  # Blue for 'credit policy 1', red for 'credit␣
 ↪policy 0'
# Plot histograms with custom colors
sns.histplot(df[df['credit.policy'] == 1]['fico'], alpha=0.5, bins=30,␣
 ↪color=colors[0], label='Credit Policy = 1')
sns.histplot(df[df['credit.policy'] == 0]['fico'], alpha=0.5, bins=30,␣
 ↪color=colors[1], label='Credit Policy = 0')
# Add legend and labels
plt.legend()
plt.xlabel('FICO', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.title('Distribution of FICO Scores by Credit Policy', fontsize=16)
# Show plot
plt.show()
```




[28]:
```
plt.figure(figsize=(10, 6))
# Define custom color palette
colors = ["#1f77b4", "#d62728"]  # Blue for 'not fully paid 1', red for 'not␣
 ↪fully paid 0'
# Plot histograms with custom colors
sns.histplot(df[df['not.fully.paid'] == 1]['fico'], alpha=0.5, bins=30,␣
 ↪color=colors[0], label='Not Fully Paid = 1')
sns.histplot(df[df['not.fully.paid'] == 0]['fico'], alpha=0.5, bins=30,␣
 ↪color=colors[1], label='Not Fully Paid = 0')
# Add legend and labels
plt.legend()
plt.xlabel('FICO', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.title('Distribution of FICO Scores by Loan Status', fontsize=16)
# Show plot
plt.show()
```

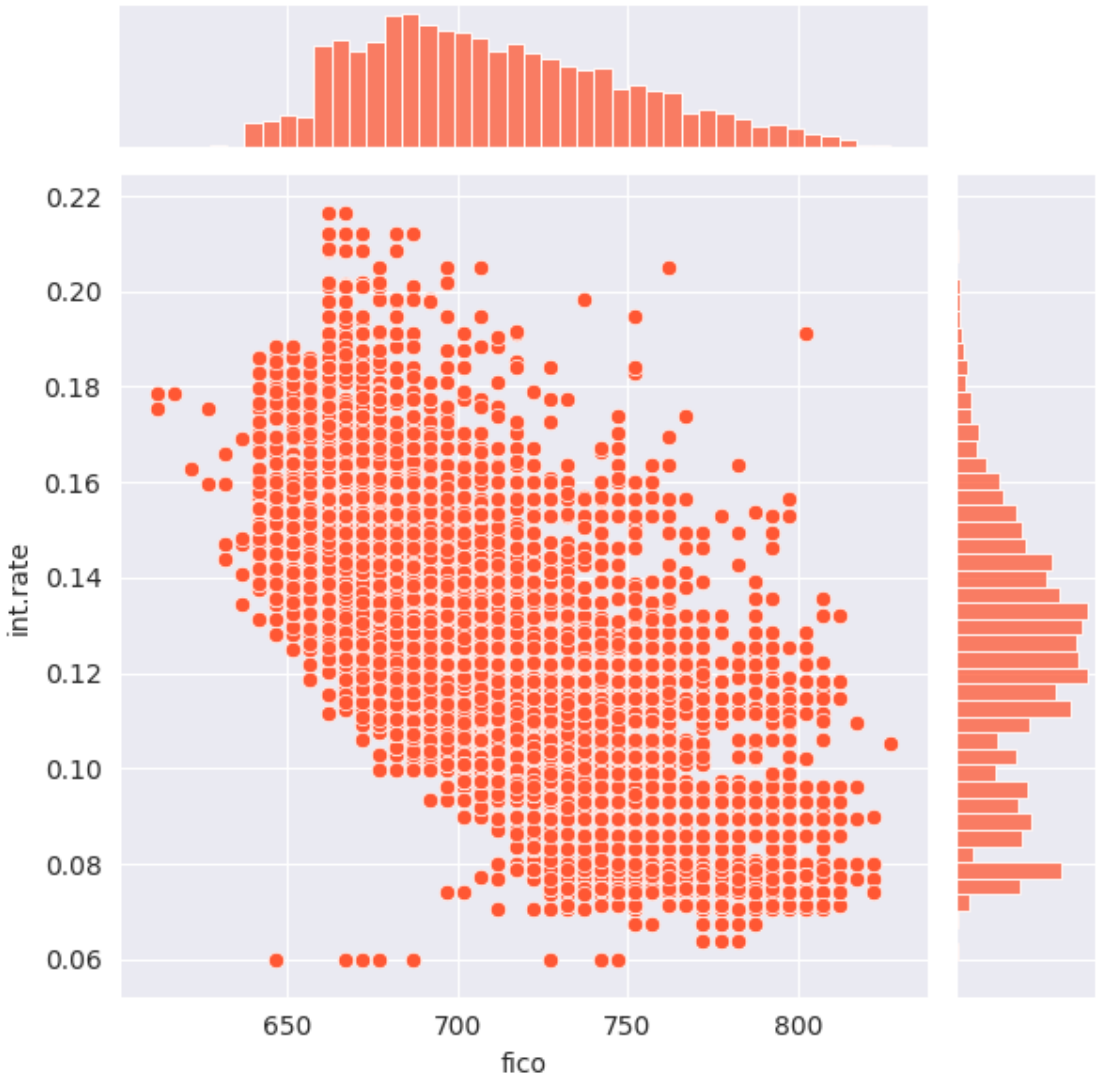



[32]:
```
plt.figure(figsize=(11, 7))
# Define custom color palette
custom_palette = ["#1f77b4", "#ff7f0e"]  # Blue for 'fully paid', orange for␣
 ↪'not fully paid'
# Plot countplot with custom color palette
sns.countplot(x='purpose', hue='not.fully.paid', data=df,␣
 ↪palette=custom_palette)
# Add legend and labels
plt.legend(title='Not Fully Paid', labels=['No', 'Yes'], fontsize=12)
plt.xlabel('Purpose', fontsize=14)
plt.ylabel('Count', fontsize=14)
plt.title('Count of Loans by Purpose and Loan Status', fontsize=16)
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')
# Show plot
plt.show()
```




Trend between FICO score and interest rate.

[34]:
```
# Define custom color
custom_color = "#FF5733"  # Orange-red color
# Plot jointplot with custom color
sns.jointplot(x='fico', y='int.rate', data=df, color=custom_color)
# Show plot
plt.show()
```




Comparing the trend between not.fully.paid and credit.policy

[36]:
```
plt.figure(figsize=(11, 7))
# Define custom color palette
custom_palette = ["#1f77b4", "#ff7f0e"]  # Blue for 'credit policy 1', orange␣
 ↪for 'credit policy 0'
# Plot lmplot with custom color palette
sns.lmplot(y='int.rate', x='fico', data=df, hue='credit.policy', col='not.fully.
 ↪paid', palette=custom_palette)
# Show plot
plt.show()
```

```
<Figure size 1100x700 with 0 Axes>
```


[37]:
```
cat_feats = ['purpose']
#cat_feats =df_test_over
```

[38]:
```
#final_data = pd.get_dummies(df,columns=cat_feats,drop_first=True)
final_data = pd.get_dummies(df_test_over,columns=cat_feats,drop_first=True)
```

[39]:
```
final_data.info()
final_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16090 entries, 0 to 9168
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  bool
 14  purpose_debt_consolidation    16090 non-null  bool
 15  purpose_educational           16090 non-null  bool
```

```
 16  purpose_home_improvement    16090 non-null  bool
 17  purpose_major_purchase      16090 non-null  bool
 18  purpose_small_business      16090 non-null  bool
dtypes: bool(6), float64(6), int64(7)
memory usage: 1.8 MB
```

[39]:

| | 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_credit_card | purpose_debt_consolidation | \ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | False | True | |
| 1 | 0 | 0 | True | False | |
| 2 | 0 | 0 | False | True | |
| 3 | 0 | 0 | False | True | |
| 4 | 0 | 0 | True | False | |

| | purpose_educational | purpose_home_improvement | purpose_major_purchase | \ |
|---|---|---|---|---|
| 0 | False | False | False | |
| 1 | False | False | False | |
| 2 | False | False | False | |
| 3 | False | False | False | |
| 4 | False | False | False | |

| | purpose_small_business |
|---|---|
| 0 | False |
| 1 | False |
| 2 | False |
| 3 | False |
| 4 | False |

[40]:
```
final_data.corr()
```

[40]:

| | credit.policy | int.rate | installment | \ |
|---|---|---|---|---|
| credit.policy | 1.000000 | -0.293307 | 0.054640 | |
| int.rate | -0.293307 | 1.000000 | 0.278189 | |
| installment | 0.054640 | 0.278189 | 1.000000 | |

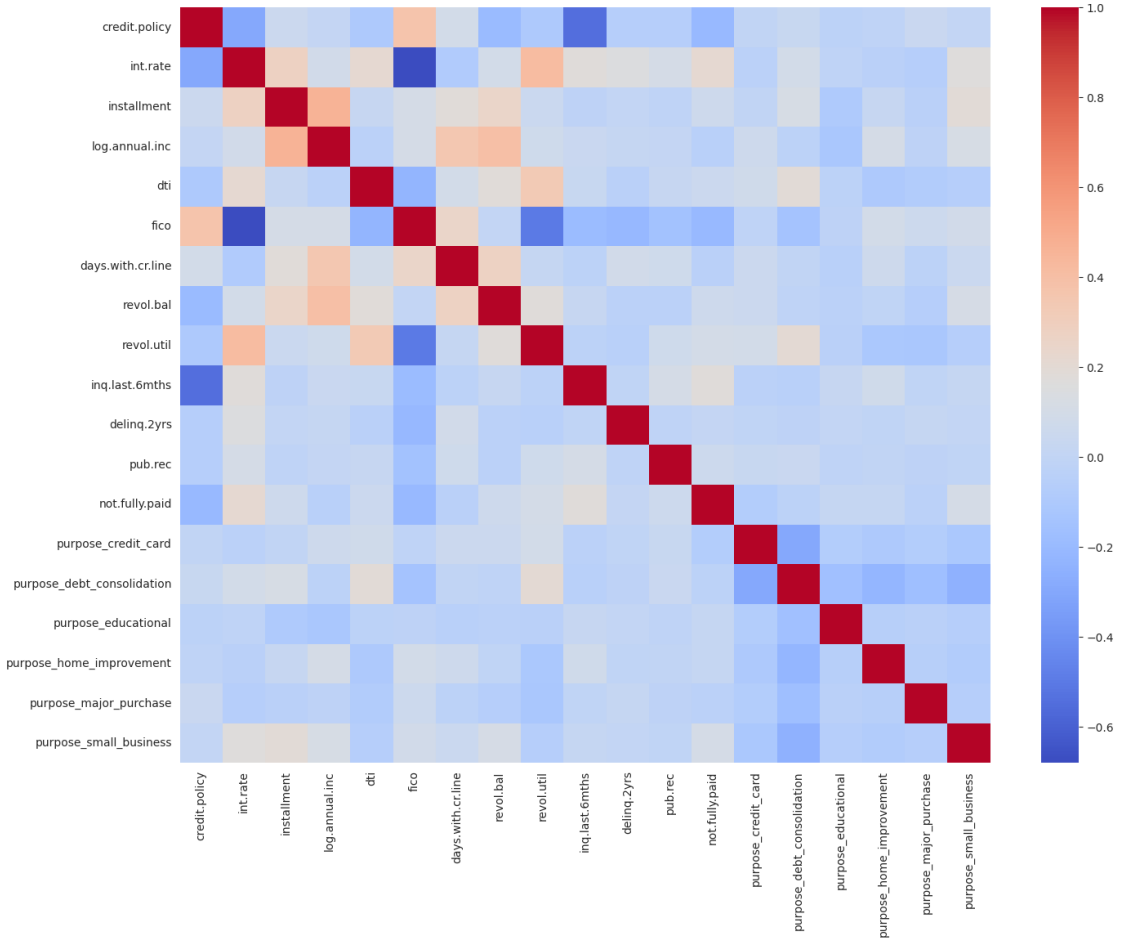
```
log.annual.inc                  0.015233  0.084073     0.468074
dti                            -0.100002  0.216350     0.024560
fico                            0.375445 -0.680607     0.106589
days.with.cr.line               0.093294 -0.088866     0.182262
revol.bal                      -0.195852  0.092401     0.249130
revol.util                     -0.099267  0.416130     0.044233
inq.last.6mths                 -0.543976  0.177614    -0.019721
delinq.2yrs                    -0.057024  0.151214     0.006644
pub.rec                        -0.060886  0.102938    -0.014513
not.fully.paid                 -0.204004  0.216851     0.061509
purpose_credit_card            -0.003451 -0.037207    -0.001608
purpose_debt_consolidation      0.030923  0.091180     0.120068
purpose_educational            -0.027396 -0.011997    -0.094194
purpose_home_improvement       -0.013367 -0.042956     0.027896
purpose_major_purchase          0.038187 -0.063104    -0.042785
purpose_small_business          0.005608  0.172527     0.186539

                           log.annual.inc       dti      fico  \
credit.policy                    0.015233 -0.100002  0.375445
int.rate                         0.084073  0.216350 -0.680607
installment                      0.468074  0.024560  0.106589
log.annual.inc                   1.000000 -0.030976  0.103173
dti                             -0.030976  1.000000 -0.230083
fico                             0.103173 -0.230083  1.000000
days.with.cr.line                0.353339  0.092904  0.247852
revol.bal                        0.400856  0.185282  0.006925
revol.util                       0.070744  0.332305 -0.497563
inq.last.6mths                   0.040748  0.030601 -0.188257
delinq.2yrs                      0.018934 -0.037275 -0.210286
pub.rec                          0.014799  0.023293 -0.154301
not.fully.paid                  -0.046827  0.051438 -0.204996
purpose_credit_card              0.065649  0.075008 -0.015966
purpose_debt_consolidation      -0.026501  0.191180 -0.145084
purpose_educational             -0.122233 -0.029502 -0.020339
purpose_home_improvement         0.106329 -0.099599  0.088097
purpose_major_purchase          -0.021422 -0.082615  0.059450
purpose_small_business           0.117499 -0.059445  0.081931

                           days.with.cr.line  revol.bal  revol.util  \
credit.policy                       0.093294  -0.195852   -0.099267
int.rate                           -0.088866   0.092401    0.416130
installment                         0.182262   0.249130    0.044233
log.annual.inc                      0.353339   0.400856    0.070744
dti                                 0.092904   0.185282    0.332305
fico                                0.247852   0.006925   -0.497563
days.with.cr.line                   1.000000   0.274217    0.015387
revol.bal                           0.274217   1.000000    0.173566
```

```
revol.util                          0.015387   0.173566    1.000000
inq.last.6mths                     -0.025762   0.023958   -0.025884
delinq.2yrs                         0.082814  -0.031442   -0.047520
pub.rec                             0.068572  -0.032100    0.073072
not.fully.paid                     -0.039680   0.061750    0.100440
purpose_credit_card                 0.051009   0.054039    0.090525
purpose_debt_consolidation         -0.004032  -0.014264    0.201880
purpose_educational                -0.045427  -0.036296   -0.041457
purpose_home_improvement            0.066779  -0.008855   -0.113380
purpose_major_purchase             -0.026191  -0.058081   -0.117610
purpose_small_business              0.041953   0.112905   -0.057804

                           inq.last.6mths  delinq.2yrs   pub.rec  \
credit.policy                  -0.543976    -0.057024 -0.060886
int.rate                        0.177614     0.151214  0.102938
installment                    -0.019721     0.006644 -0.014513
log.annual.inc                  0.040748     0.018934  0.014799
dti                             0.030601    -0.037275  0.023293
fico                           -0.188257    -0.210286 -0.154301
days.with.cr.line              -0.025762     0.082814  0.068572
revol.bal                       0.023958    -0.031442 -0.032100
revol.util                     -0.025884    -0.047520  0.073072
inq.last.6mths                  1.000000    -0.004713  0.105382
delinq.2yrs                    -0.004713     1.000000 -0.017175
pub.rec                         0.105382    -0.017175  1.000000
not.fully.paid                  0.174974     0.011607  0.056566
purpose_credit_card            -0.034921    -0.006084  0.033033
purpose_debt_consolidation     -0.049479    -0.018629  0.041151
purpose_educational             0.022866     0.003342 -0.012640
purpose_home_improvement        0.077996    -0.008494 -0.001312
purpose_major_purchase         -0.006090     0.015656 -0.024098
purpose_small_business          0.019452     0.008367 -0.008189

                           not.fully.paid  purpose_credit_card  \
credit.policy                  -0.204004            -0.003451
int.rate                        0.216851            -0.037207
installment                     0.061509            -0.001608
log.annual.inc                 -0.046827             0.065649
dti                             0.051438             0.075008
fico                           -0.204996            -0.015966
days.with.cr.line              -0.039680             0.051009
revol.bal                       0.061750             0.054039
revol.util                      0.100440             0.090525
inq.last.6mths                  0.174974            -0.034921
delinq.2yrs                     0.011607            -0.006084
pub.rec                         0.056566             0.033033
not.fully.paid                  1.000000            -0.070200
```

```
purpose_credit_card             -0.070200            1.000000
purpose_debt_consolidation      -0.028638           -0.297858
purpose_educational              0.018623           -0.071679
purpose_home_improvement         0.018087           -0.099095
purpose_major_purchase          -0.033648           -0.075473
purpose_small_business           0.101653           -0.109613

                          purpose_debt_consolidation  purpose_educational  \
credit.policy                               0.030923            -0.027396
int.rate                                    0.091180            -0.011997
installment                                 0.120068            -0.094194
log.annual.inc                             -0.026501            -0.122233
dti                                         0.191180            -0.029502
fico                                       -0.145084            -0.020339
days.with.cr.line                          -0.004032            -0.045427
revol.bal                                  -0.014264            -0.036296
revol.util                                  0.201880            -0.041457
inq.last.6mths                             -0.049479             0.022866
delinq.2yrs                                -0.018629             0.003342
pub.rec                                     0.041151            -0.012640
not.fully.paid                             -0.028638             0.018623
purpose_credit_card                        -0.297858            -0.071679
purpose_debt_consolidation                  1.000000            -0.162353
purpose_educational                        -0.162353             1.000000
purpose_home_improvement                   -0.224450            -0.054014
purpose_major_purchase                     -0.170945            -0.041138
purpose_small_business                     -0.248273            -0.059747

                          purpose_home_improvement  purpose_major_purchase  \
credit.policy                            -0.013367                0.038187
int.rate                                 -0.042956               -0.063104
installment                               0.027896               -0.042785
log.annual.inc                            0.106329               -0.021422
dti                                      -0.099599               -0.082615
fico                                      0.088097                0.059450
days.with.cr.line                         0.066779               -0.026191
revol.bal                                -0.008855               -0.058081
revol.util                               -0.113380               -0.117610
inq.last.6mths                            0.077996               -0.006090
delinq.2yrs                              -0.008494                0.015656
pub.rec                                  -0.001312               -0.024098
not.fully.paid                            0.018087               -0.033648
purpose_credit_card                      -0.099095               -0.075473
purpose_debt_consolidation               -0.224450               -0.170945
purpose_educational                      -0.054014               -0.041138
purpose_home_improvement                  1.000000               -0.056872
purpose_major_purchase                   -0.056872                1.000000
```

```
purpose_small_business                    -0.082598                -0.062908

                                purpose_small_business
credit.policy                                 0.005608
int.rate                                      0.172527
installment                                   0.186539
log.annual.inc                                0.117499
dti                                          -0.059445
fico                                          0.081931
days.with.cr.line                             0.041953
revol.bal                                     0.112905
revol.util                                   -0.057804
inq.last.6mths                                0.019452
delinq.2yrs                                   0.008367
pub.rec                                      -0.008189
not.fully.paid                                0.101653
purpose_credit_card                          -0.109613
purpose_debt_consolidation                   -0.248273
purpose_educational                          -0.059747
purpose_home_improvement                     -0.082598
purpose_major_purchase                       -0.062908
purpose_small_business                        1.000000
```

[42]:
```
plt.figure(figsize=[16, 12])
# Define custom color palette
custom_palette = "coolwarm"
# Plot heatmap with custom color palette
sns.heatmap(data=final_data.corr(), cmap=custom_palette, annot=False, fmt='.2g')
# Show plot
plt.show()
```




[43]:
```
to_drop2 = ['revol.bal', 'days.with.cr.line', 'installment', 'revol.bal']

final_data.drop(to_drop2, axis=1, inplace=True)
#We only focus on the grids of yellow or very light green. After comparing with␣
 ↪the feature description again,  revol.bal,day.with.cr.line,installment can␣
 ↪repsent by annual income. revol.util can repsent by int.rate,
```

[44]:
```
final_data.isnull().mean()
```

[44]:
```
credit.policy              0.0
int.rate                   0.0
log.annual.inc             0.0
dti                        0.0
fico                       0.0
revol.util                 0.0
inq.last.6mths             0.0
delinq.2yrs                0.0
pub.rec                    0.0
```

```
not.fully.paid                0.0
purpose_credit_card           0.0
purpose_debt_consolidation    0.0
purpose_educational           0.0
purpose_home_improvement      0.0
purpose_major_purchase        0.0
purpose_small_business        0.0
dtype: float64
```

### Modeling-Deep Learning Implementation

[45]:
```
#to_train = df1[df1['not.fully.paid'].isin([0,1])]
#to_pred = df1[df1['not.fully.paid'] == 2]

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

[46]:
```
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)
```

[47]:
```
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[48]:
```
X_train.shape
```

[48]:
```
(11263, 15)
```

[50]:
```
model = Sequential()

model.add(
        Dense(94, activation='relu')
)

model.add(
        Dense(30, activation='relu')
)

model.add(
        Dense(15, activation='relu')
)


model.add(
```

```
        Dense(1, activation='sigmoid')
)

model.compile(
        optimizer='adam',
        loss='binary_crossentropy',
        metrics=['accuracy']
)
```

[51]:
```
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 [==============================] - 2s 10ms/step - loss: 0.6691 - accuracy:
0.6104 - val_loss: 0.6528 - val_accuracy: 0.6178
Epoch 2/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6482 - accuracy:
0.6204 - val_loss: 0.6477 - val_accuracy: 0.6190
Epoch 3/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6445 - accuracy:
0.6201 - val_loss: 0.6475 - val_accuracy: 0.6207
Epoch 4/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6416 - accuracy:
0.6265 - val_loss: 0.6462 - val_accuracy: 0.6174
Epoch 5/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6404 - accuracy:
0.6245 - val_loss: 0.6440 - val_accuracy: 0.6236
Epoch 6/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6384 - accuracy:
0.6272 - val_loss: 0.6447 - val_accuracy: 0.6196
Epoch 7/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6371 - accuracy:
0.6278 - val_loss: 0.6444 - val_accuracy: 0.6213
Epoch 8/200
```

```
44/44 [==============================] - 0s 5ms/step - loss: 0.6367 - accuracy:
0.6265 - val_loss: 0.6437 - val_accuracy: 0.6219
Epoch 9/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6350 - accuracy:
0.6307 - val_loss: 0.6427 - val_accuracy: 0.6232
Epoch 10/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6349 - accuracy:
0.6321 - val_loss: 0.6391 - val_accuracy: 0.6265
Epoch 11/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6348 - accuracy:
0.6315 - val_loss: 0.6404 - val_accuracy: 0.6246
Epoch 12/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6327 - accuracy:
0.6317 - val_loss: 0.6383 - val_accuracy: 0.6259
Epoch 13/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6320 - accuracy:
0.6354 - val_loss: 0.6374 - val_accuracy: 0.6271
Epoch 14/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6307 - accuracy:
0.6355 - val_loss: 0.6363 - val_accuracy: 0.6288
Epoch 15/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6295 - accuracy:
0.6397 - val_loss: 0.6391 - val_accuracy: 0.6267
Epoch 16/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6282 - accuracy:
0.6369 - val_loss: 0.6352 - val_accuracy: 0.6335
Epoch 17/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6272 - accuracy:
0.6401 - val_loss: 0.6353 - val_accuracy: 0.6317
Epoch 18/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6266 - accuracy:
0.6387 - val_loss: 0.6337 - val_accuracy: 0.6360
Epoch 19/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6259 - accuracy:
0.6430 - val_loss: 0.6351 - val_accuracy: 0.6310
Epoch 20/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6247 - accuracy:
0.6379 - val_loss: 0.6324 - val_accuracy: 0.6358
Epoch 21/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6234 - accuracy:
0.6435 - val_loss: 0.6312 - val_accuracy: 0.6377
Epoch 22/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6219 - accuracy:
0.6433 - val_loss: 0.6388 - val_accuracy: 0.6219
Epoch 23/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6213 - accuracy:
0.6474 - val_loss: 0.6309 - val_accuracy: 0.6360
Epoch 24/200
```

```
44/44 [==============================] - 0s 6ms/step - loss: 0.6201 - accuracy:
0.6481 - val_loss: 0.6319 - val_accuracy: 0.6389
Epoch 25/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6175 - accuracy:
0.6503 - val_loss: 0.6314 - val_accuracy: 0.6362
Epoch 26/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6200 - accuracy:
0.6467 - val_loss: 0.6326 - val_accuracy: 0.6312
Epoch 27/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6167 - accuracy:
0.6516 - val_loss: 0.6279 - val_accuracy: 0.6416
Epoch 28/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6161 - accuracy:
0.6523 - val_loss: 0.6284 - val_accuracy: 0.6406
Epoch 29/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6143 - accuracy:
0.6528 - val_loss: 0.6346 - val_accuracy: 0.6310
Epoch 30/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6142 - accuracy:
0.6544 - val_loss: 0.6270 - val_accuracy: 0.6439
Epoch 31/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6131 - accuracy:
0.6508 - val_loss: 0.6265 - val_accuracy: 0.6439
Epoch 32/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6116 - accuracy:
0.6556 - val_loss: 0.6286 - val_accuracy: 0.6406
Epoch 33/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6107 - accuracy:
0.6512 - val_loss: 0.6249 - val_accuracy: 0.6482
Epoch 34/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6085 - accuracy:
0.6591 - val_loss: 0.6240 - val_accuracy: 0.6453
Epoch 35/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6062 - accuracy:
0.6585 - val_loss: 0.6243 - val_accuracy: 0.6466
Epoch 36/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6069 - accuracy:
0.6605 - val_loss: 0.6246 - val_accuracy: 0.6443
Epoch 37/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6066 - accuracy:
0.6605 - val_loss: 0.6267 - val_accuracy: 0.6410
Epoch 38/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6052 - accuracy:
0.6560 - val_loss: 0.6294 - val_accuracy: 0.6401
Epoch 39/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6023 - accuracy:
0.6635 - val_loss: 0.6271 - val_accuracy: 0.6484
Epoch 40/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.6030 - accuracy:
0.6614 - val_loss: 0.6214 - val_accuracy: 0.6491
Epoch 41/200
44/44 [==============================] - 0s 4ms/step - loss: 0.6004 - accuracy:
0.6646 - val_loss: 0.6199 - val_accuracy: 0.6511
Epoch 42/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5994 - accuracy:
0.6681 - val_loss: 0.6205 - val_accuracy: 0.6526
Epoch 43/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5999 - accuracy:
0.6646 - val_loss: 0.6220 - val_accuracy: 0.6489
Epoch 44/200
44/44 [==============================] - 0s 3ms/step - loss: 0.5974 - accuracy:
0.6663 - val_loss: 0.6233 - val_accuracy: 0.6412
Epoch 45/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5946 - accuracy:
0.6721 - val_loss: 0.6191 - val_accuracy: 0.6576
Epoch 46/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5953 - accuracy:
0.6691 - val_loss: 0.6191 - val_accuracy: 0.6576
Epoch 47/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5969 - accuracy:
0.6674 - val_loss: 0.6207 - val_accuracy: 0.6528
Epoch 48/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5934 - accuracy:
0.6725 - val_loss: 0.6232 - val_accuracy: 0.6549
Epoch 49/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5916 - accuracy:
0.6722 - val_loss: 0.6254 - val_accuracy: 0.6368
Epoch 50/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5931 - accuracy:
0.6743 - val_loss: 0.6205 - val_accuracy: 0.6507
Epoch 51/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5893 - accuracy:
0.6764 - val_loss: 0.6201 - val_accuracy: 0.6507
Epoch 52/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5882 - accuracy:
0.6787 - val_loss: 0.6183 - val_accuracy: 0.6520
Epoch 53/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5898 - accuracy:
0.6690 - val_loss: 0.6151 - val_accuracy: 0.6532
Epoch 54/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5876 - accuracy:
0.6765 - val_loss: 0.6194 - val_accuracy: 0.6547
Epoch 55/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5859 - accuracy:
0.6808 - val_loss: 0.6263 - val_accuracy: 0.6401
Epoch 56/200
```

```
44/44 [==============================] - 0s 5ms/step - loss: 0.5883 - accuracy:
0.6722 - val_loss: 0.6183 - val_accuracy: 0.6534
Epoch 57/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5865 - accuracy:
0.6793 - val_loss: 0.6196 - val_accuracy: 0.6451
Epoch 58/200
44/44 [==============================] - 0s 3ms/step - loss: 0.5825 - accuracy:
0.6813 - val_loss: 0.6152 - val_accuracy: 0.6555
Epoch 59/200
44/44 [==============================] - 0s 3ms/step - loss: 0.5804 - accuracy:
0.6816 - val_loss: 0.6248 - val_accuracy: 0.6449
Epoch 60/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5814 - accuracy:
0.6787 - val_loss: 0.6138 - val_accuracy: 0.6582
Epoch 61/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5806 - accuracy:
0.6828 - val_loss: 0.6160 - val_accuracy: 0.6582
Epoch 62/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5769 - accuracy:
0.6885 - val_loss: 0.6164 - val_accuracy: 0.6598
Epoch 63/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5779 - accuracy:
0.6880 - val_loss: 0.6125 - val_accuracy: 0.6625
Epoch 64/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5749 - accuracy:
0.6894 - val_loss: 0.6126 - val_accuracy: 0.6592
Epoch 65/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5775 - accuracy:
0.6897 - val_loss: 0.6344 - val_accuracy: 0.6401
Epoch 66/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5769 - accuracy:
0.6870 - val_loss: 0.6099 - val_accuracy: 0.6729
Epoch 67/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5732 - accuracy:
0.6932 - val_loss: 0.6100 - val_accuracy: 0.6708
Epoch 68/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5730 - accuracy:
0.6952 - val_loss: 0.6125 - val_accuracy: 0.6735
Epoch 69/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5710 - accuracy:
0.6958 - val_loss: 0.6137 - val_accuracy: 0.6718
Epoch 70/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5729 - accuracy:
0.6909 - val_loss: 0.6162 - val_accuracy: 0.6544
Epoch 71/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5686 - accuracy:
0.6946 - val_loss: 0.6206 - val_accuracy: 0.6631
Epoch 72/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.5694 - accuracy:
0.6937 - val_loss: 0.6189 - val_accuracy: 0.6553
Epoch 73/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5682 - accuracy:
0.6955 - val_loss: 0.6066 - val_accuracy: 0.6752
Epoch 74/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5639 - accuracy:
0.7036 - val_loss: 0.6075 - val_accuracy: 0.6750
Epoch 75/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5659 - accuracy:
0.6983 - val_loss: 0.6107 - val_accuracy: 0.6615
Epoch 76/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5624 - accuracy:
0.7027 - val_loss: 0.6056 - val_accuracy: 0.6820
Epoch 77/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5608 - accuracy:
0.7029 - val_loss: 0.6072 - val_accuracy: 0.6733
Epoch 78/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5614 - accuracy:
0.7035 - val_loss: 0.6034 - val_accuracy: 0.6760
Epoch 79/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5608 - accuracy:
0.7014 - val_loss: 0.6076 - val_accuracy: 0.6822
Epoch 80/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5609 - accuracy:
0.7021 - val_loss: 0.6071 - val_accuracy: 0.6752
Epoch 81/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5593 - accuracy:
0.7048 - val_loss: 0.6037 - val_accuracy: 0.6832
Epoch 82/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5598 - accuracy:
0.7032 - val_loss: 0.6065 - val_accuracy: 0.6685
Epoch 83/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5573 - accuracy:
0.7072 - val_loss: 0.6035 - val_accuracy: 0.6754
Epoch 84/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5554 - accuracy:
0.7095 - val_loss: 0.6019 - val_accuracy: 0.6779
Epoch 85/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5587 - accuracy:
0.7021 - val_loss: 0.6073 - val_accuracy: 0.6710
Epoch 86/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5526 - accuracy:
0.7121 - val_loss: 0.6000 - val_accuracy: 0.6940
Epoch 87/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5504 - accuracy:
0.7092 - val_loss: 0.6029 - val_accuracy: 0.6689
Epoch 88/200
```

```
44/44 [==============================] - 0s 6ms/step - loss: 0.5536 - accuracy:
0.7136 - val_loss: 0.6044 - val_accuracy: 0.6766
Epoch 89/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5562 - accuracy:
0.7048 - val_loss: 0.6003 - val_accuracy: 0.6849
Epoch 90/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5483 - accuracy:
0.7151 - val_loss: 0.6024 - val_accuracy: 0.6729
Epoch 91/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5528 - accuracy:
0.7089 - val_loss: 0.5991 - val_accuracy: 0.6805
Epoch 92/200
44/44 [==============================] - 0s 7ms/step - loss: 0.5486 - accuracy:
0.7150 - val_loss: 0.6042 - val_accuracy: 0.6743
Epoch 93/200
44/44 [==============================] - 0s 7ms/step - loss: 0.5480 - accuracy:
0.7155 - val_loss: 0.5993 - val_accuracy: 0.6824
Epoch 94/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5485 - accuracy:
0.7130 - val_loss: 0.6001 - val_accuracy: 0.6801
Epoch 95/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5474 - accuracy:
0.7171 - val_loss: 0.6137 - val_accuracy: 0.6702
Epoch 96/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5449 - accuracy:
0.7163 - val_loss: 0.5993 - val_accuracy: 0.6824
Epoch 97/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5414 - accuracy:
0.7201 - val_loss: 0.5989 - val_accuracy: 0.6805
Epoch 98/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5408 - accuracy:
0.7200 - val_loss: 0.6023 - val_accuracy: 0.6787
Epoch 99/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5403 - accuracy:
0.7198 - val_loss: 0.6041 - val_accuracy: 0.6762
Epoch 100/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5415 - accuracy:
0.7177 - val_loss: 0.6069 - val_accuracy: 0.6801
Epoch 101/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5393 - accuracy:
0.7202 - val_loss: 0.5991 - val_accuracy: 0.6855
Epoch 102/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5379 - accuracy:
0.7207 - val_loss: 0.6009 - val_accuracy: 0.6799
Epoch 103/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5418 - accuracy:
0.7184 - val_loss: 0.5939 - val_accuracy: 0.6928
Epoch 104/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.5414 - accuracy:
0.7205 - val_loss: 0.6168 - val_accuracy: 0.6677
Epoch 105/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5367 - accuracy:
0.7242 - val_loss: 0.6005 - val_accuracy: 0.6866
Epoch 106/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5338 - accuracy:
0.7246 - val_loss: 0.6004 - val_accuracy: 0.6859
Epoch 107/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5318 - accuracy:
0.7291 - val_loss: 0.6096 - val_accuracy: 0.6729
Epoch 108/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5343 - accuracy:
0.7299 - val_loss: 0.5945 - val_accuracy: 0.6870
Epoch 109/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5320 - accuracy:
0.7301 - val_loss: 0.6008 - val_accuracy: 0.6940
Epoch 110/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5307 - accuracy:
0.7258 - val_loss: 0.5996 - val_accuracy: 0.6853
Epoch 111/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5285 - accuracy:
0.7281 - val_loss: 0.6014 - val_accuracy: 0.6766
Epoch 112/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5265 - accuracy:
0.7308 - val_loss: 0.5957 - val_accuracy: 0.6930
Epoch 113/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5258 - accuracy:
0.7302 - val_loss: 0.5991 - val_accuracy: 0.6895
Epoch 114/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5286 - accuracy:
0.7284 - val_loss: 0.5928 - val_accuracy: 0.6863
Epoch 115/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5266 - accuracy:
0.7276 - val_loss: 0.5916 - val_accuracy: 0.6942
Epoch 116/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5264 - accuracy:
0.7318 - val_loss: 0.5965 - val_accuracy: 0.6793
Epoch 117/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5259 - accuracy:
0.7290 - val_loss: 0.5907 - val_accuracy: 0.6961
Epoch 118/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5226 - accuracy:
0.7342 - val_loss: 0.5899 - val_accuracy: 0.6961
Epoch 119/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5268 - accuracy:
0.7286 - val_loss: 0.5969 - val_accuracy: 0.6928
Epoch 120/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.5243 - accuracy:
0.7288 - val_loss: 0.5947 - val_accuracy: 0.6888
Epoch 121/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5262 - accuracy:
0.7304 - val_loss: 0.6011 - val_accuracy: 0.6783
Epoch 122/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5216 - accuracy:
0.7291 - val_loss: 0.5975 - val_accuracy: 0.6901
Epoch 123/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5220 - accuracy:
0.7312 - val_loss: 0.5871 - val_accuracy: 0.7044
Epoch 124/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5183 - accuracy:
0.7319 - val_loss: 0.5928 - val_accuracy: 0.6973
Epoch 125/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5179 - accuracy:
0.7394 - val_loss: 0.5966 - val_accuracy: 0.6834
Epoch 126/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5172 - accuracy:
0.7335 - val_loss: 0.5960 - val_accuracy: 0.6834
Epoch 127/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5178 - accuracy:
0.7344 - val_loss: 0.5928 - val_accuracy: 0.6853
Epoch 128/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5153 - accuracy:
0.7397 - val_loss: 0.5914 - val_accuracy: 0.6971
Epoch 129/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5134 - accuracy:
0.7391 - val_loss: 0.5900 - val_accuracy: 0.6979
Epoch 130/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5134 - accuracy:
0.7401 - val_loss: 0.5992 - val_accuracy: 0.6888
Epoch 131/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5192 - accuracy:
0.7312 - val_loss: 0.5954 - val_accuracy: 0.6866
Epoch 132/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5113 - accuracy:
0.7403 - val_loss: 0.5943 - val_accuracy: 0.7064
Epoch 133/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5153 - accuracy:
0.7348 - val_loss: 0.5908 - val_accuracy: 0.7023
Epoch 134/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5108 - accuracy:
0.7404 - val_loss: 0.5879 - val_accuracy: 0.6996
Epoch 135/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5095 - accuracy:
0.7436 - val_loss: 0.5917 - val_accuracy: 0.7023
Epoch 136/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.5091 - accuracy:
0.7421 - val_loss: 0.5869 - val_accuracy: 0.7004
Epoch 137/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5097 - accuracy:
0.7417 - val_loss: 0.6028 - val_accuracy: 0.6965
Epoch 138/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5125 - accuracy:
0.7394 - val_loss: 0.5952 - val_accuracy: 0.6874
Epoch 139/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5070 - accuracy:
0.7421 - val_loss: 0.5912 - val_accuracy: 0.6994
Epoch 140/200
44/44 [==============================] - 0s 4ms/step - loss: 0.5069 - accuracy:
0.7442 - val_loss: 0.5868 - val_accuracy: 0.7037
Epoch 141/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5082 - accuracy:
0.7398 - val_loss: 0.6034 - val_accuracy: 0.6963
Epoch 142/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5047 - accuracy:
0.7397 - val_loss: 0.6091 - val_accuracy: 0.6822
Epoch 143/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5047 - accuracy:
0.7453 - val_loss: 0.5953 - val_accuracy: 0.7029
Epoch 144/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5030 - accuracy:
0.7460 - val_loss: 0.5874 - val_accuracy: 0.6969
Epoch 145/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5047 - accuracy:
0.7454 - val_loss: 0.5898 - val_accuracy: 0.7048
Epoch 146/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5070 - accuracy:
0.7382 - val_loss: 0.5829 - val_accuracy: 0.7021
Epoch 147/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5030 - accuracy:
0.7464 - val_loss: 0.5896 - val_accuracy: 0.7013
Epoch 148/200
44/44 [==============================] - 0s 6ms/step - loss: 0.4971 - accuracy:
0.7490 - val_loss: 0.5857 - val_accuracy: 0.6973
Epoch 149/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5002 - accuracy:
0.7451 - val_loss: 0.5901 - val_accuracy: 0.7062
Epoch 150/200
44/44 [==============================] - 0s 6ms/step - loss: 0.4987 - accuracy:
0.7452 - val_loss: 0.5853 - val_accuracy: 0.6971
Epoch 151/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5011 - accuracy:
0.7480 - val_loss: 0.5914 - val_accuracy: 0.6934
Epoch 152/200
```

```
44/44 [==============================] - 0s 6ms/step - loss: 0.5003 - accuracy:
0.7468 - val_loss: 0.6108 - val_accuracy: 0.6843
Epoch 153/200
44/44 [==============================] - 0s 6ms/step - loss: 0.4968 - accuracy:
0.7463 - val_loss: 0.5882 - val_accuracy: 0.6953
Epoch 154/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4933 - accuracy:
0.7508 - val_loss: 0.5863 - val_accuracy: 0.7011
Epoch 155/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5001 - accuracy:
0.7462 - val_loss: 0.5852 - val_accuracy: 0.7067
Epoch 156/200
44/44 [==============================] - 0s 6ms/step - loss: 0.4919 - accuracy:
0.7502 - val_loss: 0.5836 - val_accuracy: 0.7091
Epoch 157/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4929 - accuracy:
0.7542 - val_loss: 0.5848 - val_accuracy: 0.7029
Epoch 158/200
44/44 [==============================] - 0s 6ms/step - loss: 0.4941 - accuracy:
0.7497 - val_loss: 0.5927 - val_accuracy: 0.7008
Epoch 159/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4925 - accuracy:
0.7523 - val_loss: 0.5838 - val_accuracy: 0.7098
Epoch 160/200
44/44 [==============================] - 0s 3ms/step - loss: 0.4933 - accuracy:
0.7485 - val_loss: 0.5926 - val_accuracy: 0.6975
Epoch 161/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4915 - accuracy:
0.7504 - val_loss: 0.6096 - val_accuracy: 0.6892
Epoch 162/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4946 - accuracy:
0.7506 - val_loss: 0.5883 - val_accuracy: 0.7071
Epoch 163/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4880 - accuracy:
0.7558 - val_loss: 0.6010 - val_accuracy: 0.6909
Epoch 164/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4890 - accuracy:
0.7504 - val_loss: 0.5887 - val_accuracy: 0.6924
Epoch 165/200
44/44 [==============================] - 0s 3ms/step - loss: 0.4854 - accuracy:
0.7518 - val_loss: 0.5828 - val_accuracy: 0.7079
Epoch 166/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4877 - accuracy:
0.7549 - val_loss: 0.6079 - val_accuracy: 0.6847
Epoch 167/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4862 - accuracy:
0.7551 - val_loss: 0.5817 - val_accuracy: 0.7015
Epoch 168/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.4861 - accuracy:
0.7514 - val_loss: 0.5861 - val_accuracy: 0.6992
Epoch 169/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4843 - accuracy:
0.7583 - val_loss: 0.5856 - val_accuracy: 0.7075
Epoch 170/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4866 - accuracy:
0.7538 - val_loss: 0.5910 - val_accuracy: 0.6965
Epoch 171/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4851 - accuracy:
0.7545 - val_loss: 0.5938 - val_accuracy: 0.6984
Epoch 172/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4859 - accuracy:
0.7545 - val_loss: 0.5867 - val_accuracy: 0.7035
Epoch 173/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4881 - accuracy:
0.7510 - val_loss: 0.5837 - val_accuracy: 0.7027
Epoch 174/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4822 - accuracy:
0.7599 - val_loss: 0.5907 - val_accuracy: 0.7035
Epoch 175/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4853 - accuracy:
0.7568 - val_loss: 0.5832 - val_accuracy: 0.7060
Epoch 176/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4762 - accuracy:
0.7643 - val_loss: 0.5862 - val_accuracy: 0.6988
Epoch 177/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4842 - accuracy:
0.7580 - val_loss: 0.5825 - val_accuracy: 0.7023
Epoch 178/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4823 - accuracy:
0.7557 - val_loss: 0.5988 - val_accuracy: 0.7058
Epoch 179/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4778 - accuracy:
0.7596 - val_loss: 0.6027 - val_accuracy: 0.6905
Epoch 180/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4783 - accuracy:
0.7591 - val_loss: 0.5846 - val_accuracy: 0.7021
Epoch 181/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4799 - accuracy:
0.7579 - val_loss: 0.5909 - val_accuracy: 0.7004
Epoch 182/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4747 - accuracy:
0.7637 - val_loss: 0.5949 - val_accuracy: 0.6984
Epoch 183/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4769 - accuracy:
0.7612 - val_loss: 0.5885 - val_accuracy: 0.6979
Epoch 184/200
```

```
44/44 [==============================] - 0s 4ms/step - loss: 0.4756 - accuracy:
0.7593 - val_loss: 0.5792 - val_accuracy: 0.7108
Epoch 185/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4745 - accuracy:
0.7624 - val_loss: 0.5845 - val_accuracy: 0.6969
Epoch 186/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4789 - accuracy:
0.7566 - val_loss: 0.6021 - val_accuracy: 0.7040
Epoch 187/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4730 - accuracy:
0.7662 - val_loss: 0.5880 - val_accuracy: 0.7073
Epoch 188/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4785 - accuracy:
0.7590 - val_loss: 0.5961 - val_accuracy: 0.6953
Epoch 189/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4845 - accuracy:
0.7490 - val_loss: 0.6006 - val_accuracy: 0.7006
Epoch 190/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4741 - accuracy:
0.7623 - val_loss: 0.5866 - val_accuracy: 0.7075
Epoch 191/200
44/44 [==============================] - 0s 3ms/step - loss: 0.4724 - accuracy:
0.7657 - val_loss: 0.5803 - val_accuracy: 0.7127
Epoch 192/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4724 - accuracy:
0.7628 - val_loss: 0.5858 - val_accuracy: 0.7050
Epoch 193/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4694 - accuracy:
0.7646 - val_loss: 0.5824 - val_accuracy: 0.7222
Epoch 194/200
44/44 [==============================] - 0s 5ms/step - loss: 0.4660 - accuracy:
0.7716 - val_loss: 0.5791 - val_accuracy: 0.7108
Epoch 195/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4687 - accuracy:
0.7697 - val_loss: 0.5863 - val_accuracy: 0.7052
Epoch 196/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4712 - accuracy:
0.7644 - val_loss: 0.5862 - val_accuracy: 0.7091
Epoch 197/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4667 - accuracy:
0.7665 - val_loss: 0.5943 - val_accuracy: 0.7050
Epoch 198/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4655 - accuracy:
0.7692 - val_loss: 0.5877 - val_accuracy: 0.7096
Epoch 199/200
44/44 [==============================] - 0s 4ms/step - loss: 0.4715 - accuracy:
0.7616 - val_loss: 0.6229 - val_accuracy: 0.6984
Epoch 200/200
```

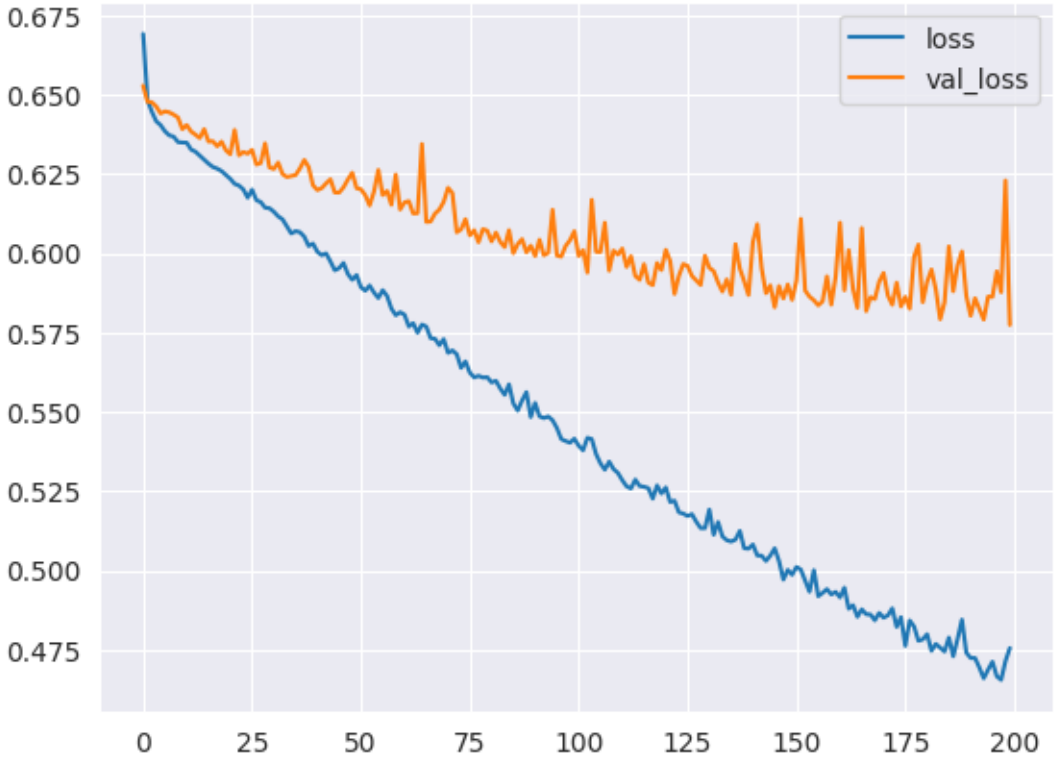
```
44/44 [==============================] - 0s 4ms/step - loss: 0.4756 - accuracy:
0.7586 - val_loss: 0.5774 - val_accuracy: 0.7222
```

[51]:
```
<keras.src.callbacks.History at 0x7db75d309120>
```

### Model Evaluation and Validation

[52]:
```
pd.DataFrame(model.history.history)[['loss','val_loss']].plot() #over fitting
```

[52]:
```
<Axes: >
```




[56]:
```
predictions = model.predict(X_test)
predicted_classes = (predictions > 0.5).astype('int32')

print(
    confusion_matrix(y_test, predicted_classes),
    '\n',
    classification_report(y_test, predicted_classes)
)
```

```
151/151 [==============================] - 0s 1ms/step
[[1779  658]
 [ 683 1707]]
```

```
                precision    recall  f1-score   support

           0       0.72      0.73      0.73      2437
           1       0.72      0.71      0.72      2390

    accuracy                           0.72      4827
   macro avg       0.72      0.72      0.72      4827
weighted avg       0.72      0.72      0.72      4827
```

The model achieved an overall accuracy of approximately 72%, with a precision of 72% and recall of 73% for class 0 (not fully paid), and a precision of 72% and recall of 71% for class 1 (fully paid). Despite a balanced performance between the two classes, there were 683 false negatives, indicating room for improvement in correctly identifying fully paid loans."

**Model Refinement**

[58]:
```
model_new = Sequential()

model_new.add(
        Dense(94, activation='relu')
)

model_new.add(Dropout(0.2))

model_new.add(
        Dense(30, activation='relu')
)

model_new.add(Dropout(0.2))

model_new.add(
        Dense(15, 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']
)


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 [==============================] - 2s 20ms/step - loss: 0.6824 -
binary_accuracy: 0.5579 - val_loss: 0.6623 - val_binary_accuracy: 0.6145
Epoch 2/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6634 -
binary_accuracy: 0.6033 - val_loss: 0.6541 - val_binary_accuracy: 0.6136
Epoch 3/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6577 -
binary_accuracy: 0.6151 - val_loss: 0.6518 - val_binary_accuracy: 0.6124
Epoch 4/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6541 -
binary_accuracy: 0.6159 - val_loss: 0.6504 - val_binary_accuracy: 0.6066
Epoch 5/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6518 -
binary_accuracy: 0.6162 - val_loss: 0.6477 - val_binary_accuracy: 0.6219
Epoch 6/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6508 -
binary_accuracy: 0.6208 - val_loss: 0.6462 - val_binary_accuracy: 0.6217
Epoch 7/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6495 -
binary_accuracy: 0.6186 - val_loss: 0.6455 - val_binary_accuracy: 0.6153
Epoch 8/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6486 -
binary_accuracy: 0.6178 - val_loss: 0.6443 - val_binary_accuracy: 0.6230
Epoch 9/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6462 -
binary_accuracy: 0.6196 - val_loss: 0.6443 - val_binary_accuracy: 0.6221
Epoch 10/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6459 -
binary_accuracy: 0.6214 - val_loss: 0.6429 - val_binary_accuracy: 0.6246
Epoch 11/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6446 -
binary_accuracy: 0.6240 - val_loss: 0.6420 - val_binary_accuracy: 0.6215
Epoch 12/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6443 -
binary_accuracy: 0.6210 - val_loss: 0.6415 - val_binary_accuracy: 0.6207
Epoch 13/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6428 -
binary_accuracy: 0.6213 - val_loss: 0.6408 - val_binary_accuracy: 0.6238
```

```
Epoch 14/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6447 -
binary_accuracy: 0.6162 - val_loss: 0.6414 - val_binary_accuracy: 0.6269
Epoch 15/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6406 -
binary_accuracy: 0.6270 - val_loss: 0.6397 - val_binary_accuracy: 0.6267
Epoch 16/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6402 -
binary_accuracy: 0.6251 - val_loss: 0.6391 - val_binary_accuracy: 0.6248
Epoch 17/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6398 -
binary_accuracy: 0.6282 - val_loss: 0.6397 - val_binary_accuracy: 0.6242
Epoch 18/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6395 -
binary_accuracy: 0.6253 - val_loss: 0.6386 - val_binary_accuracy: 0.6298
Epoch 19/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6386 -
binary_accuracy: 0.6267 - val_loss: 0.6381 - val_binary_accuracy: 0.6254
Epoch 20/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6380 -
binary_accuracy: 0.6261 - val_loss: 0.6374 - val_binary_accuracy: 0.6310
Epoch 21/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6372 -
binary_accuracy: 0.6290 - val_loss: 0.6379 - val_binary_accuracy: 0.6254
Epoch 22/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6372 -
binary_accuracy: 0.6283 - val_loss: 0.6382 - val_binary_accuracy: 0.6271
Epoch 23/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6376 -
binary_accuracy: 0.6239 - val_loss: 0.6359 - val_binary_accuracy: 0.6285
Epoch 24/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6363 -
binary_accuracy: 0.6304 - val_loss: 0.6384 - val_binary_accuracy: 0.6242
Epoch 25/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6367 -
binary_accuracy: 0.6273 - val_loss: 0.6356 - val_binary_accuracy: 0.6327
Epoch 26/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6357 -
binary_accuracy: 0.6329 - val_loss: 0.6356 - val_binary_accuracy: 0.6300
Epoch 27/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6351 -
binary_accuracy: 0.6321 - val_loss: 0.6343 - val_binary_accuracy: 0.6356
Epoch 28/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6341 -
binary_accuracy: 0.6317 - val_loss: 0.6382 - val_binary_accuracy: 0.6343
Epoch 29/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6344 -
binary_accuracy: 0.6299 - val_loss: 0.6337 - val_binary_accuracy: 0.6377
```

```
Epoch 30/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6328 -
binary_accuracy: 0.6349 - val_loss: 0.6339 - val_binary_accuracy: 0.6385
Epoch 31/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6343 -
binary_accuracy: 0.6326 - val_loss: 0.6327 - val_binary_accuracy: 0.6335
Epoch 32/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6316 -
binary_accuracy: 0.6362 - val_loss: 0.6327 - val_binary_accuracy: 0.6366
Epoch 33/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6327 -
binary_accuracy: 0.6323 - val_loss: 0.6329 - val_binary_accuracy: 0.6333
Epoch 34/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6307 -
binary_accuracy: 0.6330 - val_loss: 0.6319 - val_binary_accuracy: 0.6335
Epoch 35/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6316 -
binary_accuracy: 0.6375 - val_loss: 0.6313 - val_binary_accuracy: 0.6385
Epoch 36/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6291 -
binary_accuracy: 0.6361 - val_loss: 0.6303 - val_binary_accuracy: 0.6379
Epoch 37/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6292 -
binary_accuracy: 0.6359 - val_loss: 0.6301 - val_binary_accuracy: 0.6389
Epoch 38/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6283 -
binary_accuracy: 0.6399 - val_loss: 0.6298 - val_binary_accuracy: 0.6439
Epoch 39/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6301 -
binary_accuracy: 0.6322 - val_loss: 0.6298 - val_binary_accuracy: 0.6383
Epoch 40/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6275 -
binary_accuracy: 0.6369 - val_loss: 0.6295 - val_binary_accuracy: 0.6401
Epoch 41/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6266 -
binary_accuracy: 0.6430 - val_loss: 0.6290 - val_binary_accuracy: 0.6366
Epoch 42/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6274 -
binary_accuracy: 0.6409 - val_loss: 0.6283 - val_binary_accuracy: 0.6437
Epoch 43/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6254 -
binary_accuracy: 0.6461 - val_loss: 0.6285 - val_binary_accuracy: 0.6412
Epoch 44/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6276 -
binary_accuracy: 0.6441 - val_loss: 0.6281 - val_binary_accuracy: 0.6468
Epoch 45/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6262 -
binary_accuracy: 0.6423 - val_loss: 0.6276 - val_binary_accuracy: 0.6451
```

```
Epoch 46/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6239 -
binary_accuracy: 0.6412 - val_loss: 0.6266 - val_binary_accuracy: 0.6414
Epoch 47/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6238 -
binary_accuracy: 0.6461 - val_loss: 0.6266 - val_binary_accuracy: 0.6453
Epoch 48/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6233 -
binary_accuracy: 0.6460 - val_loss: 0.6261 - val_binary_accuracy: 0.6412
Epoch 49/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6225 -
binary_accuracy: 0.6459 - val_loss: 0.6260 - val_binary_accuracy: 0.6418
Epoch 50/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6236 -
binary_accuracy: 0.6463 - val_loss: 0.6255 - val_binary_accuracy: 0.6449
Epoch 51/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6218 -
binary_accuracy: 0.6487 - val_loss: 0.6252 - val_binary_accuracy: 0.6430
Epoch 52/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6214 -
binary_accuracy: 0.6501 - val_loss: 0.6246 - val_binary_accuracy: 0.6453
Epoch 53/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6235 -
binary_accuracy: 0.6504 - val_loss: 0.6239 - val_binary_accuracy: 0.6493
Epoch 54/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6203 -
binary_accuracy: 0.6452 - val_loss: 0.6243 - val_binary_accuracy: 0.6453
Epoch 55/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6229 -
binary_accuracy: 0.6449 - val_loss: 0.6256 - val_binary_accuracy: 0.6408
Epoch 56/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6182 -
binary_accuracy: 0.6512 - val_loss: 0.6249 - val_binary_accuracy: 0.6418
Epoch 57/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6192 -
binary_accuracy: 0.6552 - val_loss: 0.6240 - val_binary_accuracy: 0.6507
Epoch 58/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6196 -
binary_accuracy: 0.6479 - val_loss: 0.6225 - val_binary_accuracy: 0.6497
Epoch 59/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6171 -
binary_accuracy: 0.6540 - val_loss: 0.6235 - val_binary_accuracy: 0.6443
Epoch 60/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6183 -
binary_accuracy: 0.6542 - val_loss: 0.6217 - val_binary_accuracy: 0.6486
Epoch 61/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6186 -
binary_accuracy: 0.6536 - val_loss: 0.6205 - val_binary_accuracy: 0.6530
```

```
Epoch 62/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6155 -
binary_accuracy: 0.6532 - val_loss: 0.6205 - val_binary_accuracy: 0.6511
Epoch 63/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6141 -
binary_accuracy: 0.6566 - val_loss: 0.6219 - val_binary_accuracy: 0.6482
Epoch 64/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6158 -
binary_accuracy: 0.6504 - val_loss: 0.6196 - val_binary_accuracy: 0.6528
Epoch 65/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6158 -
binary_accuracy: 0.6534 - val_loss: 0.6210 - val_binary_accuracy: 0.6493
Epoch 66/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6148 -
binary_accuracy: 0.6584 - val_loss: 0.6204 - val_binary_accuracy: 0.6518
Epoch 67/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6160 -
binary_accuracy: 0.6604 - val_loss: 0.6211 - val_binary_accuracy: 0.6497
Epoch 68/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6129 -
binary_accuracy: 0.6543 - val_loss: 0.6218 - val_binary_accuracy: 0.6493
Epoch 69/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6128 -
binary_accuracy: 0.6565 - val_loss: 0.6233 - val_binary_accuracy: 0.6422
Epoch 70/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6134 -
binary_accuracy: 0.6555 - val_loss: 0.6186 - val_binary_accuracy: 0.6478
Epoch 71/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6119 -
binary_accuracy: 0.6543 - val_loss: 0.6171 - val_binary_accuracy: 0.6561
Epoch 72/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6116 -
binary_accuracy: 0.6578 - val_loss: 0.6181 - val_binary_accuracy: 0.6542
Epoch 73/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6083 -
binary_accuracy: 0.6623 - val_loss: 0.6172 - val_binary_accuracy: 0.6553
Epoch 74/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6097 -
binary_accuracy: 0.6556 - val_loss: 0.6184 - val_binary_accuracy: 0.6528
Epoch 75/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6093 -
binary_accuracy: 0.6594 - val_loss: 0.6167 - val_binary_accuracy: 0.6567
Epoch 76/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6054 -
binary_accuracy: 0.6628 - val_loss: 0.6172 - val_binary_accuracy: 0.6592
Epoch 77/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6093 -
binary_accuracy: 0.6611 - val_loss: 0.6147 - val_binary_accuracy: 0.6619
```

```
Epoch 78/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6080 -
binary_accuracy: 0.6612 - val_loss: 0.6157 - val_binary_accuracy: 0.6588
Epoch 79/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6069 -
binary_accuracy: 0.6602 - val_loss: 0.6148 - val_binary_accuracy: 0.6555
Epoch 80/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6059 -
binary_accuracy: 0.6610 - val_loss: 0.6185 - val_binary_accuracy: 0.6530
Epoch 81/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6056 -
binary_accuracy: 0.6623 - val_loss: 0.6148 - val_binary_accuracy: 0.6561
Epoch 82/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6051 -
binary_accuracy: 0.6622 - val_loss: 0.6132 - val_binary_accuracy: 0.6619
Epoch 83/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6033 -
binary_accuracy: 0.6661 - val_loss: 0.6123 - val_binary_accuracy: 0.6638
Epoch 84/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6043 -
binary_accuracy: 0.6646 - val_loss: 0.6145 - val_binary_accuracy: 0.6544
Epoch 85/200
44/44 [==============================] - 0s 7ms/step - loss: 0.6029 -
binary_accuracy: 0.6634 - val_loss: 0.6120 - val_binary_accuracy: 0.6654
Epoch 86/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6048 -
binary_accuracy: 0.6661 - val_loss: 0.6120 - val_binary_accuracy: 0.6584
Epoch 87/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6030 -
binary_accuracy: 0.6643 - val_loss: 0.6114 - val_binary_accuracy: 0.6631
Epoch 88/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6038 -
binary_accuracy: 0.6648 - val_loss: 0.6104 - val_binary_accuracy: 0.6634
Epoch 89/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6018 -
binary_accuracy: 0.6664 - val_loss: 0.6104 - val_binary_accuracy: 0.6631
Epoch 90/200
44/44 [==============================] - 0s 10ms/step - loss: 0.6025 -
binary_accuracy: 0.6647 - val_loss: 0.6095 - val_binary_accuracy: 0.6660
Epoch 91/200
44/44 [==============================] - 0s 8ms/step - loss: 0.6024 -
binary_accuracy: 0.6600 - val_loss: 0.6111 - val_binary_accuracy: 0.6663
Epoch 92/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6011 -
binary_accuracy: 0.6620 - val_loss: 0.6157 - val_binary_accuracy: 0.6532
Epoch 93/200
44/44 [==============================] - 0s 5ms/step - loss: 0.6017 -
binary_accuracy: 0.6697 - val_loss: 0.6103 - val_binary_accuracy: 0.6634
```

```
Epoch 94/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5998 -
binary_accuracy: 0.6728 - val_loss: 0.6084 - val_binary_accuracy: 0.6673
Epoch 95/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5995 -
binary_accuracy: 0.6694 - val_loss: 0.6073 - val_binary_accuracy: 0.6650
Epoch 96/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5989 -
binary_accuracy: 0.6670 - val_loss: 0.6115 - val_binary_accuracy: 0.6640
Epoch 97/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5985 -
binary_accuracy: 0.6682 - val_loss: 0.6092 - val_binary_accuracy: 0.6648
Epoch 98/200
44/44 [==============================] - 0s 6ms/step - loss: 0.6005 -
binary_accuracy: 0.6649 - val_loss: 0.6081 - val_binary_accuracy: 0.6636
Epoch 99/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5985 -
binary_accuracy: 0.6663 - val_loss: 0.6066 - val_binary_accuracy: 0.6660
Epoch 100/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5957 -
binary_accuracy: 0.6713 - val_loss: 0.6051 - val_binary_accuracy: 0.6650
Epoch 101/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5954 -
binary_accuracy: 0.6722 - val_loss: 0.6050 - val_binary_accuracy: 0.6698
Epoch 102/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5962 -
binary_accuracy: 0.6730 - val_loss: 0.6053 - val_binary_accuracy: 0.6696
Epoch 103/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5964 -
binary_accuracy: 0.6706 - val_loss: 0.6071 - val_binary_accuracy: 0.6613
Epoch 104/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5930 -
binary_accuracy: 0.6739 - val_loss: 0.6052 - val_binary_accuracy: 0.6660
Epoch 105/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5932 -
binary_accuracy: 0.6742 - val_loss: 0.6066 - val_binary_accuracy: 0.6638
Epoch 106/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5904 -
binary_accuracy: 0.6774 - val_loss: 0.6034 - val_binary_accuracy: 0.6714
Epoch 107/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5932 -
binary_accuracy: 0.6734 - val_loss: 0.6028 - val_binary_accuracy: 0.6708
Epoch 108/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5886 -
binary_accuracy: 0.6782 - val_loss: 0.6045 - val_binary_accuracy: 0.6694
Epoch 109/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5888 -
binary_accuracy: 0.6806 - val_loss: 0.5995 - val_binary_accuracy: 0.6747
```

```
Epoch 110/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5916 -
binary_accuracy: 0.6746 - val_loss: 0.6011 - val_binary_accuracy: 0.6687
Epoch 111/200
44/44 [==============================] - 1s 16ms/step - loss: 0.5837 -
binary_accuracy: 0.6814 - val_loss: 0.6015 - val_binary_accuracy: 0.6758
Epoch 112/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5900 -
binary_accuracy: 0.6777 - val_loss: 0.6013 - val_binary_accuracy: 0.6704
Epoch 113/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5884 -
binary_accuracy: 0.6780 - val_loss: 0.6012 - val_binary_accuracy: 0.6681
Epoch 114/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5877 -
binary_accuracy: 0.6797 - val_loss: 0.6035 - val_binary_accuracy: 0.6687
Epoch 115/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5913 -
binary_accuracy: 0.6738 - val_loss: 0.6015 - val_binary_accuracy: 0.6675
Epoch 116/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5900 -
binary_accuracy: 0.6764 - val_loss: 0.5989 - val_binary_accuracy: 0.6708
Epoch 117/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5878 -
binary_accuracy: 0.6796 - val_loss: 0.5987 - val_binary_accuracy: 0.6721
Epoch 118/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5861 -
binary_accuracy: 0.6797 - val_loss: 0.5996 - val_binary_accuracy: 0.6685
Epoch 119/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5854 -
binary_accuracy: 0.6800 - val_loss: 0.5977 - val_binary_accuracy: 0.6731
Epoch 120/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5837 -
binary_accuracy: 0.6789 - val_loss: 0.5979 - val_binary_accuracy: 0.6739
Epoch 121/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5842 -
binary_accuracy: 0.6811 - val_loss: 0.5964 - val_binary_accuracy: 0.6708
Epoch 122/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5886 -
binary_accuracy: 0.6779 - val_loss: 0.5944 - val_binary_accuracy: 0.6789
Epoch 123/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5837 -
binary_accuracy: 0.6829 - val_loss: 0.5945 - val_binary_accuracy: 0.6743
Epoch 124/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5876 -
binary_accuracy: 0.6801 - val_loss: 0.5948 - val_binary_accuracy: 0.6760
Epoch 125/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5848 -
binary_accuracy: 0.6809 - val_loss: 0.5930 - val_binary_accuracy: 0.6694
```

```
Epoch 126/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5790 -
binary_accuracy: 0.6891 - val_loss: 0.5944 - val_binary_accuracy: 0.6758
Epoch 127/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5841 -
binary_accuracy: 0.6811 - val_loss: 0.5947 - val_binary_accuracy: 0.6721
Epoch 128/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5811 -
binary_accuracy: 0.6827 - val_loss: 0.5953 - val_binary_accuracy: 0.6731
Epoch 129/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5809 -
binary_accuracy: 0.6873 - val_loss: 0.5923 - val_binary_accuracy: 0.6770
Epoch 130/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5797 -
binary_accuracy: 0.6859 - val_loss: 0.5937 - val_binary_accuracy: 0.6731
Epoch 131/200
44/44 [==============================] - 0s 9ms/step - loss: 0.5810 -
binary_accuracy: 0.6865 - val_loss: 0.5914 - val_binary_accuracy: 0.6764
Epoch 132/200
44/44 [==============================] - 0s 9ms/step - loss: 0.5817 -
binary_accuracy: 0.6843 - val_loss: 0.5951 - val_binary_accuracy: 0.6710
Epoch 133/200
44/44 [==============================] - 0s 10ms/step - loss: 0.5796 -
binary_accuracy: 0.6843 - val_loss: 0.5931 - val_binary_accuracy: 0.6756
Epoch 134/200
44/44 [==============================] - 0s 9ms/step - loss: 0.5809 -
binary_accuracy: 0.6860 - val_loss: 0.5913 - val_binary_accuracy: 0.6737
Epoch 135/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5763 -
binary_accuracy: 0.6883 - val_loss: 0.5935 - val_binary_accuracy: 0.6795
Epoch 136/200
44/44 [==============================] - 0s 9ms/step - loss: 0.5781 -
binary_accuracy: 0.6848 - val_loss: 0.5909 - val_binary_accuracy: 0.6770
Epoch 137/200
44/44 [==============================] - 0s 7ms/step - loss: 0.5758 -
binary_accuracy: 0.6872 - val_loss: 0.5907 - val_binary_accuracy: 0.6793
Epoch 138/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5786 -
binary_accuracy: 0.6851 - val_loss: 0.5918 - val_binary_accuracy: 0.6764
Epoch 139/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5769 -
binary_accuracy: 0.6893 - val_loss: 0.5886 - val_binary_accuracy: 0.6801
Epoch 140/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5757 -
binary_accuracy: 0.6865 - val_loss: 0.5883 - val_binary_accuracy: 0.6805
Epoch 141/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5756 -
binary_accuracy: 0.6872 - val_loss: 0.5873 - val_binary_accuracy: 0.6772
```

```
Epoch 142/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5753 -
binary_accuracy: 0.6871 - val_loss: 0.5903 - val_binary_accuracy: 0.6772
Epoch 143/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5726 -
binary_accuracy: 0.6893 - val_loss: 0.5883 - val_binary_accuracy: 0.6810
Epoch 144/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5706 -
binary_accuracy: 0.6912 - val_loss: 0.5890 - val_binary_accuracy: 0.6768
Epoch 145/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5665 -
binary_accuracy: 0.6928 - val_loss: 0.5889 - val_binary_accuracy: 0.6789
Epoch 146/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5707 -
binary_accuracy: 0.6900 - val_loss: 0.5901 - val_binary_accuracy: 0.6783
Epoch 147/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5734 -
binary_accuracy: 0.6852 - val_loss: 0.5868 - val_binary_accuracy: 0.6795
Epoch 148/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5710 -
binary_accuracy: 0.6933 - val_loss: 0.5886 - val_binary_accuracy: 0.6832
Epoch 149/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5723 -
binary_accuracy: 0.6900 - val_loss: 0.5882 - val_binary_accuracy: 0.6895
Epoch 150/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5716 -
binary_accuracy: 0.6881 - val_loss: 0.5921 - val_binary_accuracy: 0.6826
Epoch 151/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5692 -
binary_accuracy: 0.6939 - val_loss: 0.5852 - val_binary_accuracy: 0.6899
Epoch 152/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5701 -
binary_accuracy: 0.6975 - val_loss: 0.5865 - val_binary_accuracy: 0.6870
Epoch 153/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5715 -
binary_accuracy: 0.6956 - val_loss: 0.5854 - val_binary_accuracy: 0.6812
Epoch 154/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5705 -
binary_accuracy: 0.6923 - val_loss: 0.5835 - val_binary_accuracy: 0.6882
Epoch 155/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5662 -
binary_accuracy: 0.6990 - val_loss: 0.5815 - val_binary_accuracy: 0.6950
Epoch 156/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5661 -
binary_accuracy: 0.6926 - val_loss: 0.5839 - val_binary_accuracy: 0.6897
Epoch 157/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5682 -
binary_accuracy: 0.6975 - val_loss: 0.5834 - val_binary_accuracy: 0.6892
```

```
Epoch 158/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5682 -
binary_accuracy: 0.6946 - val_loss: 0.5825 - val_binary_accuracy: 0.6915
Epoch 159/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5633 -
binary_accuracy: 0.6975 - val_loss: 0.5824 - val_binary_accuracy: 0.6890
Epoch 160/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5657 -
binary_accuracy: 0.6940 - val_loss: 0.5825 - val_binary_accuracy: 0.6897
Epoch 161/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5643 -
binary_accuracy: 0.6960 - val_loss: 0.5819 - val_binary_accuracy: 0.6872
Epoch 162/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5654 -
binary_accuracy: 0.6995 - val_loss: 0.5832 - val_binary_accuracy: 0.6921
Epoch 163/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5679 -
binary_accuracy: 0.6940 - val_loss: 0.5826 - val_binary_accuracy: 0.6884
Epoch 164/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5655 -
binary_accuracy: 0.6995 - val_loss: 0.5813 - val_binary_accuracy: 0.6909
Epoch 165/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5594 -
binary_accuracy: 0.7043 - val_loss: 0.5810 - val_binary_accuracy: 0.6938
Epoch 166/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5658 -
binary_accuracy: 0.6931 - val_loss: 0.5818 - val_binary_accuracy: 0.6903
Epoch 167/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5672 -
binary_accuracy: 0.6981 - val_loss: 0.5808 - val_binary_accuracy: 0.6876
Epoch 168/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5625 -
binary_accuracy: 0.7004 - val_loss: 0.5797 - val_binary_accuracy: 0.6928
Epoch 169/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5625 -
binary_accuracy: 0.7019 - val_loss: 0.5850 - val_binary_accuracy: 0.6907
Epoch 170/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5585 -
binary_accuracy: 0.7059 - val_loss: 0.5786 - val_binary_accuracy: 0.6942
Epoch 171/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5650 -
binary_accuracy: 0.7017 - val_loss: 0.5782 - val_binary_accuracy: 0.6936
Epoch 172/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5632 -
binary_accuracy: 0.6950 - val_loss: 0.5821 - val_binary_accuracy: 0.6921
Epoch 173/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5600 -
binary_accuracy: 0.7017 - val_loss: 0.5787 - val_binary_accuracy: 0.6959
```

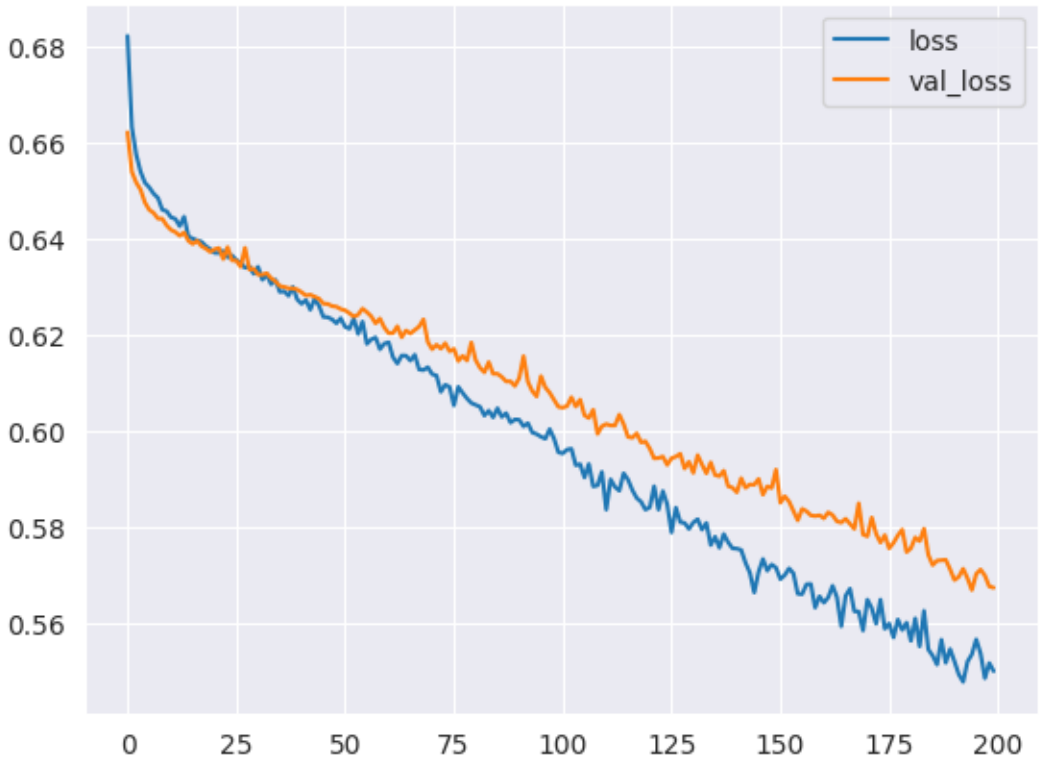
```
Epoch 174/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5650 -
binary_accuracy: 0.6975 - val_loss: 0.5769 - val_binary_accuracy: 0.6998
Epoch 175/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5589 -
binary_accuracy: 0.7077 - val_loss: 0.5784 - val_binary_accuracy: 0.6971
Epoch 176/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5600 -
binary_accuracy: 0.7003 - val_loss: 0.5757 - val_binary_accuracy: 0.7002
Epoch 177/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5571 -
binary_accuracy: 0.7027 - val_loss: 0.5768 - val_binary_accuracy: 0.6915
Epoch 178/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5608 -
binary_accuracy: 0.6973 - val_loss: 0.5783 - val_binary_accuracy: 0.6942
Epoch 179/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5587 -
binary_accuracy: 0.6975 - val_loss: 0.5796 - val_binary_accuracy: 0.6930
Epoch 180/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5602 -
binary_accuracy: 0.7035 - val_loss: 0.5749 - val_binary_accuracy: 0.7058
Epoch 181/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5564 -
binary_accuracy: 0.7022 - val_loss: 0.5757 - val_binary_accuracy: 0.6998
Epoch 182/200
44/44 [==============================] - 0s 7ms/step - loss: 0.5611 -
binary_accuracy: 0.7049 - val_loss: 0.5779 - val_binary_accuracy: 0.6946
Epoch 183/200
44/44 [==============================] - 0s 9ms/step - loss: 0.5553 -
binary_accuracy: 0.7032 - val_loss: 0.5772 - val_binary_accuracy: 0.6953
Epoch 184/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5626 -
binary_accuracy: 0.7035 - val_loss: 0.5797 - val_binary_accuracy: 0.6955
Epoch 185/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5546 -
binary_accuracy: 0.7053 - val_loss: 0.5743 - val_binary_accuracy: 0.6984
Epoch 186/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5533 -
binary_accuracy: 0.7051 - val_loss: 0.5722 - val_binary_accuracy: 0.6986
Epoch 187/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5514 -
binary_accuracy: 0.7068 - val_loss: 0.5731 - val_binary_accuracy: 0.6994
Epoch 188/200
44/44 [==============================] - 0s 8ms/step - loss: 0.5566 -
binary_accuracy: 0.7031 - val_loss: 0.5732 - val_binary_accuracy: 0.6975
Epoch 189/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5518 -
binary_accuracy: 0.7128 - val_loss: 0.5733 - val_binary_accuracy: 0.6926
```

```
Epoch 190/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5547 -
binary_accuracy: 0.7090 - val_loss: 0.5713 - val_binary_accuracy: 0.7006
Epoch 191/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5521 -
binary_accuracy: 0.7082 - val_loss: 0.5691 - val_binary_accuracy: 0.6988
Epoch 192/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5494 -
binary_accuracy: 0.7103 - val_loss: 0.5697 - val_binary_accuracy: 0.6998
Epoch 193/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5479 -
binary_accuracy: 0.7078 - val_loss: 0.5714 - val_binary_accuracy: 0.6959
Epoch 194/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5521 -
binary_accuracy: 0.7077 - val_loss: 0.5694 - val_binary_accuracy: 0.7052
Epoch 195/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5536 -
binary_accuracy: 0.7072 - val_loss: 0.5670 - val_binary_accuracy: 0.7031
Epoch 196/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5567 -
binary_accuracy: 0.7031 - val_loss: 0.5703 - val_binary_accuracy: 0.7081
Epoch 197/200
44/44 [==============================] - 0s 6ms/step - loss: 0.5538 -
binary_accuracy: 0.7074 - val_loss: 0.5712 - val_binary_accuracy: 0.6998
Epoch 198/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5486 -
binary_accuracy: 0.7105 - val_loss: 0.5700 - val_binary_accuracy: 0.7058
Epoch 199/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5518 -
binary_accuracy: 0.7098 - val_loss: 0.5677 - val_binary_accuracy: 0.7027
Epoch 200/200
44/44 [==============================] - 0s 5ms/step - loss: 0.5501 -
binary_accuracy: 0.7076 - val_loss: 0.5675 - val_binary_accuracy: 0.7050
```

[58]:
```
<keras.src.callbacks.History at 0x7db75d1d9fc0>
```

[59]:
```
pd.DataFrame(model_new.history.history)[['loss','val_loss']].plot() #The graph␣
↪shows that, by adding in Dropout layers, we have reduced the overfitting␣
↪issue compared with the old model
```

[59]:
```
<Axes: >
```




Compared to the old model, overfitting issue have been reduced by adding in dropout layers

[61]:
```
predictions_new = (model_new.predict(X_test) >= 0.2).astype('int')

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

```
151/151 [==============================] - 1s 5ms/step
[[ 490 1947]
 [  31 2359]]
              precision    recall  f1-score   support

           0       0.94      0.20      0.33      2437
           1       0.55      0.99      0.70      2390

    accuracy                           0.59      4827
   macro avg       0.74      0.59      0.52      4827
weighted avg       0.75      0.59      0.52      4827
```

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

**Saving the model and scalar.**

[62]:
```
dump(scaler, open('scaler.pkl', 'wb'))
model_new.save('my_model_lending_club.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. This
file format is considered legacy. We recommend using instead the native Keras
format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

**Model Use Case**

[63]:
```
later_scaler = load(open('scaler.pkl', 'rb'))
later_model = load_model('my_model_lending_club.h5')
```

[64]:
```
X_OOT = to_pred.drop('not.fully.paid', axis=1).values
to_pred.drop('not.fully.paid', axis=1).values

print(X_OOT.shape)
```

```
(0, 15)
```