In [37]:

```
# import libraries
import os
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
```

In [38]:

```
loan = pd.read_csv('C:/Users/User/Downloads/sda/AI/step 4/loan_data.csv',encoding = "ISO-
8859-1", low_memory=False)
```

In [39]:

```
loan.shape
```

Out[39]:

```
(9578, 14)
```

In [40]:

```
loan.describe()
```

Out[40]:

	credit.policy	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util
count	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9.578000e+03	9578.000000
mean	0.804970	0.122640	319.089413	10.932117	12.606679	710.846314	4560.767197	1.691396e+04	46.799236
std	0.396245	0.026847	207.071301	0.614813	6.883970	37.970537	2496.930377	3.375619e+04	29.014417
min	0.000000	0.060000	15.670000	7.547502	0.000000	612.000000	178.958333	0.000000e+00	0.000000
25%	1.000000	0.103900	163.770000	10.558414	7.212500	682.000000	2820.000000	3.187000e+03	22.600000
50%	1.000000	0.122100	268.950000	10.928884	12.665000	707.000000	4139.958333	8.596000e+03	46.300000
75%	1.000000	0.140700	432.762500	11.291293	17.950000	737.000000	5730.000000	1.824950e+04	70.900000
max	1.000000	0.216400	940.140000	14.528354	29.960000	827.000000	17639.958330	1.207359e+06	119.000000

In [41]:

```
loan.head(10)
```

Out[41]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6n
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	

2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5
5	1	credit_card	0.0788	125.13	11.904968	16.98	727	6120.041667	50807	51.0
6	1	debt_consolidation	0.1496	194.02	10.714418	4.00	667	3180.041667	3839	76.8
7	1	all_other	0.1114	131.22	11.002100	11.08	722	5116.000000	24220	68.6
8	1	home_improvement	0.1134	87.19	11.407565	17.25	682	3989.000000	69909	51.1
9	1	debt_consolidation	0.1221	84.12	10.203592	10.00	707	2730.041667	5630	23.0

In [42]:

```
loan.dtypes
```

Out[42]:

```
credit.policy          int64
purpose               object
int.rate             float64
installment          float64
log.annual.inc       float64
dti                  float64
fico                   int64
days.with.cr.line    float64
revol.bal              int64
revol.util           float64
inq.last.6mths         int64
delinq.2yrs            int64
pub.rec                int64
not.fully.paid         int64
dtype: object
```

In [43]:

```
#Transform categorical values into numerical values
obj_loan = loan.select_dtypes(include=['object']).copy()
obj_loan.head()
```

Out[43]:

	purpose
0	debt_consolidation
1	credit_card
2	debt_consolidation
3	debt_consolidation
4	credit_card

In [44]:

```
#obj_loan[obj_loan.isnull().any(axis=1)]
```

In [45]:

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

Out[45]:

```
debt_consolidation    3957
all_other             2331
credit_card           1262
```

```
home_improvement       629
small_business         619
major_purchase         437
educational            343
Name: purpose, dtype: int64
```

In [46]:

```
obj_loan = obj_loan.fillna({"purpose" : "credit_card"})
```

In [47]:

```
cleanup_nums = {"purpose": {"credit_card": 1,"debt_consolidation": 2 }}
```

In [48]:

```
obj_loan=obj_loan.replace(cleanup_nums)
obj_loan.head()
```

Out[48]:

	purpose
0	2
1	1
2	2
3	2
4	1

In [49]:

```
#clean_loan = loan[:]#pd.read_csv('clean_loan.csv',encoding='utf-8')
```

In [50]:

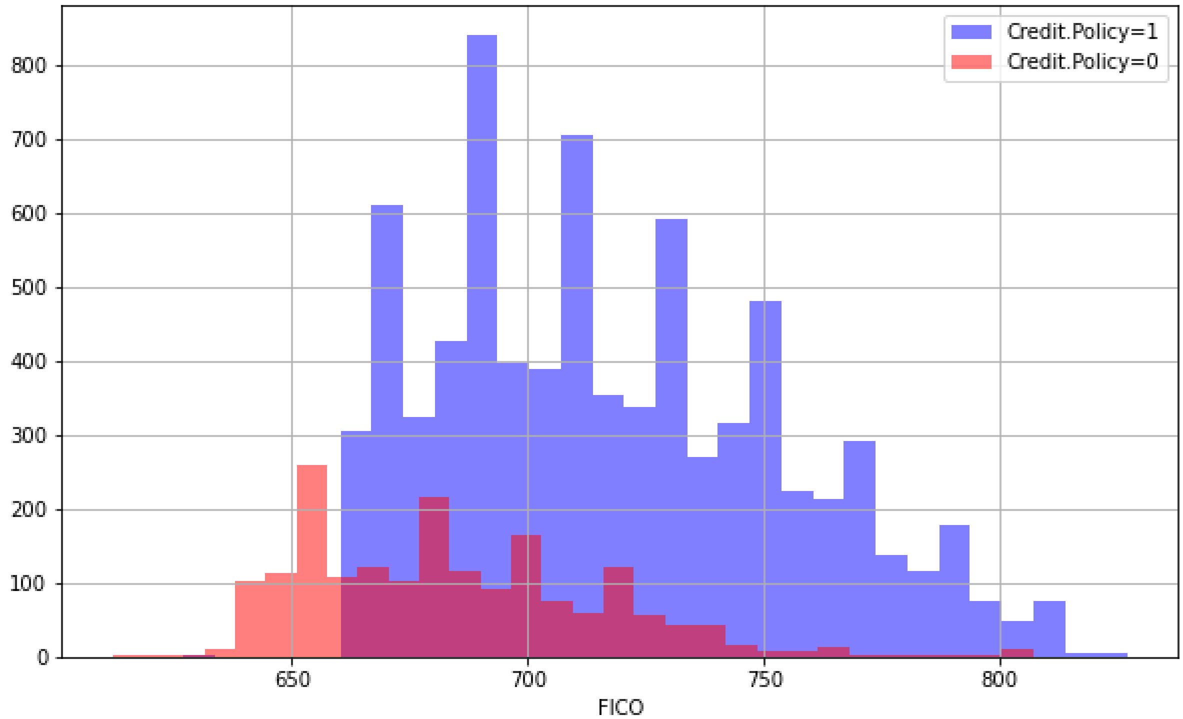
```
#clean_loan.nunique().sort_values()
```

In [51]:

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

Out[51]:

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

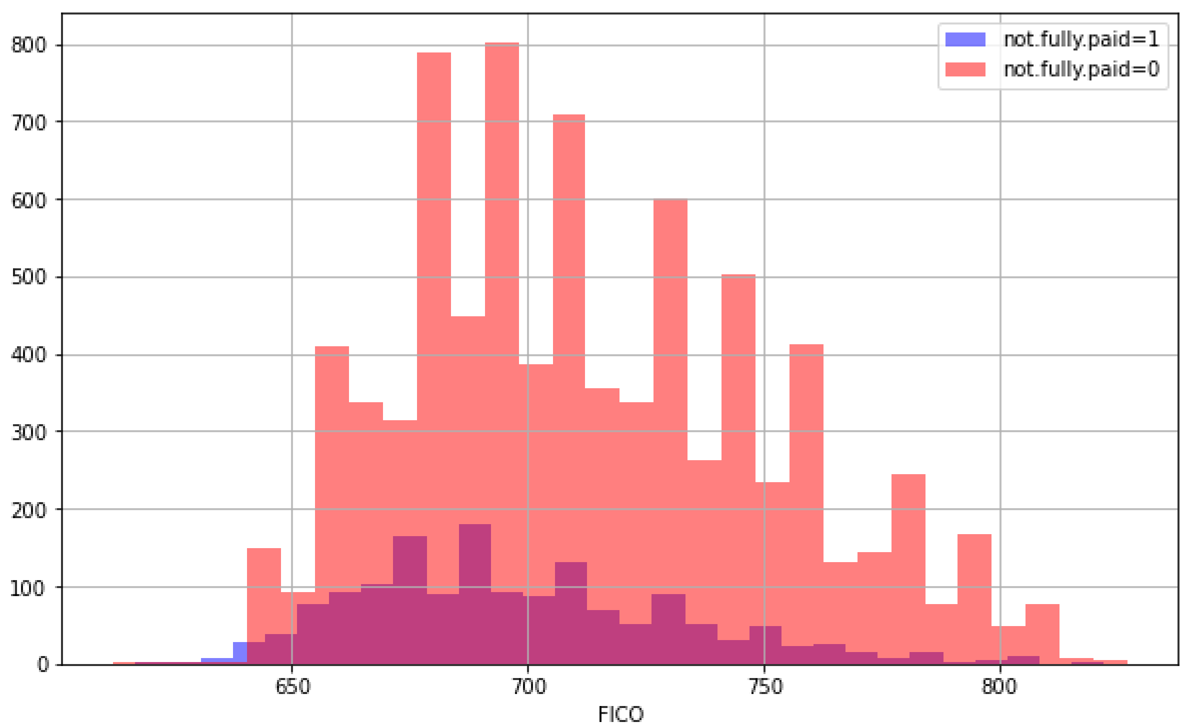


In [52]:

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

Out[52]:

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

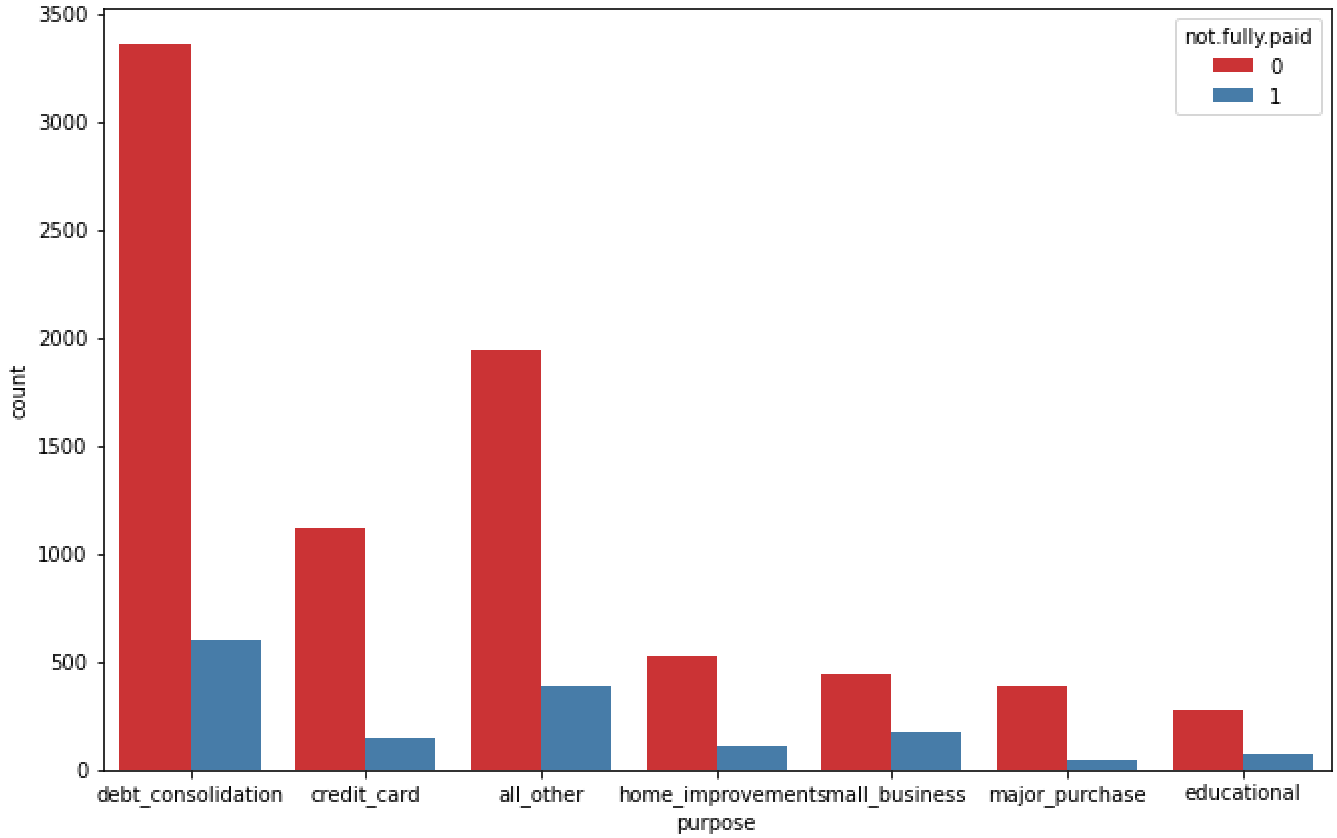


In [53]:

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

Out[53]:

```
<AxesSubplot:xlabel='purpose', ylabel='count'>
```

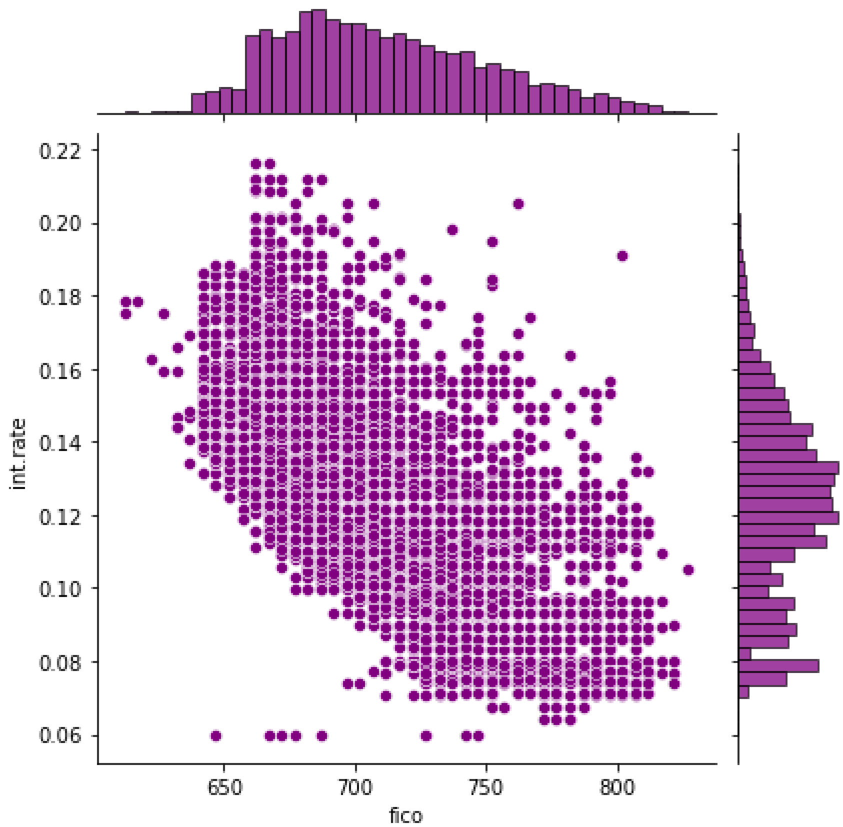


In [54]:

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

Out[54]:

```
<seaborn.axisgrid.JointGrid at 0x207b8b55cd0>
```

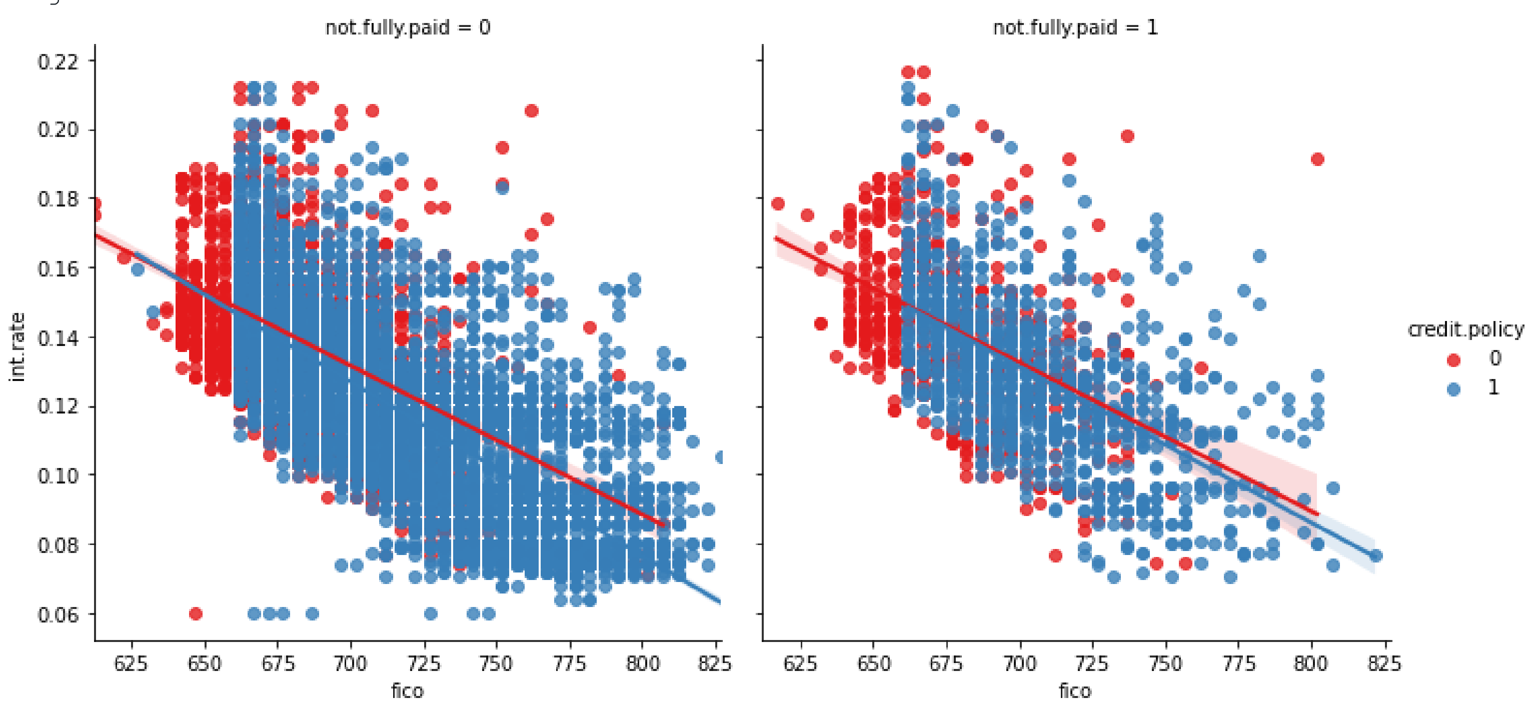


In [55]:

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

Out[55]:

```
<seaborn.axisgrid.FacetGrid at 0x207b972a160>
<Figure size 792x504 with 0 Axes>
```



In [56]:

```
loan_num = loan.select_dtypes(include = ['float64','int64'])
```

```
loan_num.head()
```

Out[56]:

	credit.policy	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	p
0	1	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	0	0	
1	1	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	0	0	
2	1	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6	1	0	
3	1	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2	1	0	
4	1	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1	

In [57]:

```
#loan_num.hist(figsize=(16,20), bins=50, xlabelsize=8, ylabelsize=8);
```

In [58]:

```
#for i in range(0, len(loan_num.columns),5):
    #sns.pairplot(data=loan_num, x_vars=loan_num.columns[i:i+5],y_vars=['log.annual.inc'])
```

In [59]:

```
#loan_num_corr = loan_num.corr()['int.rate'][:-1] # -1 because the latest row is SalePrice
#golden_features_list = loan_num_corr[abs(loan_num_corr) >
0.5].sort_values(ascending=False)
#print("There is {} strongly correlated values with
rate:\n{}".format(len(golden_features_list), golden_features_list))
```

In [60]:

```
#correlation
cor_matrix = loan.corr().abs()
print(cor_matrix)
```

```
                   credit.policy  int.rate  installment  log.annual.inc  \
credit.policy           1.000000  0.294089     0.058770        0.034906
int.rate                0.294089  1.000000     0.276140        0.056383
installment             0.058770  0.276140     1.000000        0.448102
log.annual.inc          0.034906  0.056383     0.448102        1.000000
dti                     0.090901  0.220006     0.050202        0.054065
fico                    0.348319  0.714821     0.086039        0.114576
days.with.cr.line       0.099026  0.124022     0.183297        0.336896
revol.bal               0.187518  0.092527     0.233625        0.372140
revol.util              0.104095  0.464837     0.081356        0.054881
inq.last.6mths          0.535511  0.202780     0.010419        0.029171
delinq.2yrs             0.076318  0.156079     0.004368        0.029203
pub.rec                 0.054243  0.098162     0.032760        0.016506
not.fully.paid          0.158119  0.159552     0.049955        0.033439

                        dti      fico  days.with.cr.line  revol.bal  \
credit.policy      0.090901  0.348319           0.099026   0.187518
int.rate           0.220006  0.714821           0.124022   0.092527
installment        0.050202  0.086039           0.183297   0.233625
log.annual.inc     0.054065  0.114576           0.336896   0.372140
dti                1.000000  0.241191           0.060101   0.188748
fico               0.241191  1.000000           0.263880   0.015553
days.with.cr.line  0.060101  0.263880           1.000000   0.229344
revol.bal          0.188748  0.015553           0.229344   1.000000
revol.util         0.337109  0.541289           0.024239   0.203779
inq.last.6mths     0.029189  0.185293           0.041736   0.022394
delinq.2yrs        0.021792  0.216340           0.081374   0.033243
pub.rec            0.006209  0.147592           0.071826   0.031010
not.fully.paid     0.037362  0.149666           0.029237   0.053699
```

```
                    revol.util  inq.last.6mths  delinq.2yrs   pub.rec  \
credit.policy         0.104095        0.535511     0.076318  0.054243
int.rate              0.464837        0.202780     0.156079  0.098162
installment           0.081356        0.010419     0.004368  0.032760
log.annual.inc        0.054881        0.029171     0.029203  0.016506
dti                   0.337109        0.029189     0.021792  0.006209
fico                  0.541289        0.185293     0.216340  0.147592
days.with.cr.line     0.024239        0.041736     0.081374  0.071826
revol.bal             0.203779        0.022394     0.033243  0.031010
revol.util            1.000000        0.013880     0.042740  0.066717
inq.last.6mths        0.013880        1.000000     0.021245  0.072673
delinq.2yrs           0.042740        0.021245     1.000000  0.009184
pub.rec               0.066717        0.072673     0.009184  1.000000
not.fully.paid        0.082088        0.149452     0.008881  0.048634

                    not.fully.paid
credit.policy             0.158119
int.rate                  0.159552
installment               0.049955
log.annual.inc            0.033439
dti                       0.037362
fico                      0.149666
days.with.cr.line         0.029237
revol.bal                 0.053699
revol.util                0.082088
inq.last.6mths            0.149452
delinq.2yrs               0.008881
pub.rec                   0.048634
not.fully.paid            1.000000
```

In [61]:

```
upper_tri = cor_matrix.where(np.triu(np.ones(cor_matrix.shape),k=1).astype(np.bool))
print(upper_tri)
```

```
                    credit.policy  int.rate  installment  log.annual.inc  \
credit.policy                 NaN  0.294089      0.05877        0.034906
int.rate                      NaN       NaN      0.27614        0.056383
installment                   NaN       NaN          NaN        0.448102
log.annual.inc                NaN       NaN          NaN             NaN
dti                           NaN       NaN          NaN             NaN
fico                          NaN       NaN          NaN             NaN
days.with.cr.line             NaN       NaN          NaN             NaN
revol.bal                     NaN       NaN          NaN             NaN
revol.util                    NaN       NaN          NaN             NaN
inq.last.6mths                NaN       NaN          NaN             NaN
delinq.2yrs                   NaN       NaN          NaN             NaN
pub.rec                       NaN       NaN          NaN             NaN
not.fully.paid                NaN       NaN          NaN             NaN

                         dti      fico  days.with.cr.line  revol.bal  \
credit.policy       0.090901  0.348319           0.099026   0.187518
int.rate            0.220006  0.714821           0.124022   0.092527
installment         0.050202  0.086039           0.183297   0.233625
log.annual.inc      0.054065  0.114576           0.336896   0.372140
dti                      NaN  0.241191           0.060101   0.188748
fico                     NaN       NaN           0.263880   0.015553
days.with.cr.line        NaN       NaN                NaN   0.229344
revol.bal                NaN       NaN                NaN        NaN
revol.util               NaN       NaN                NaN        NaN
inq.last.6mths           NaN       NaN                NaN        NaN
delinq.2yrs              NaN       NaN                NaN        NaN
pub.rec                  NaN       NaN                NaN        NaN
not.fully.paid           NaN       NaN                NaN        NaN

                    revol.util  inq.last.6mths  delinq.2yrs   pub.rec  \
credit.policy         0.104095        0.535511     0.076318  0.054243
int.rate              0.464837        0.202780     0.156079  0.098162
installment           0.081356        0.010419     0.004368  0.032760
log.annual.inc        0.054881        0.029171     0.029203  0.016506
```

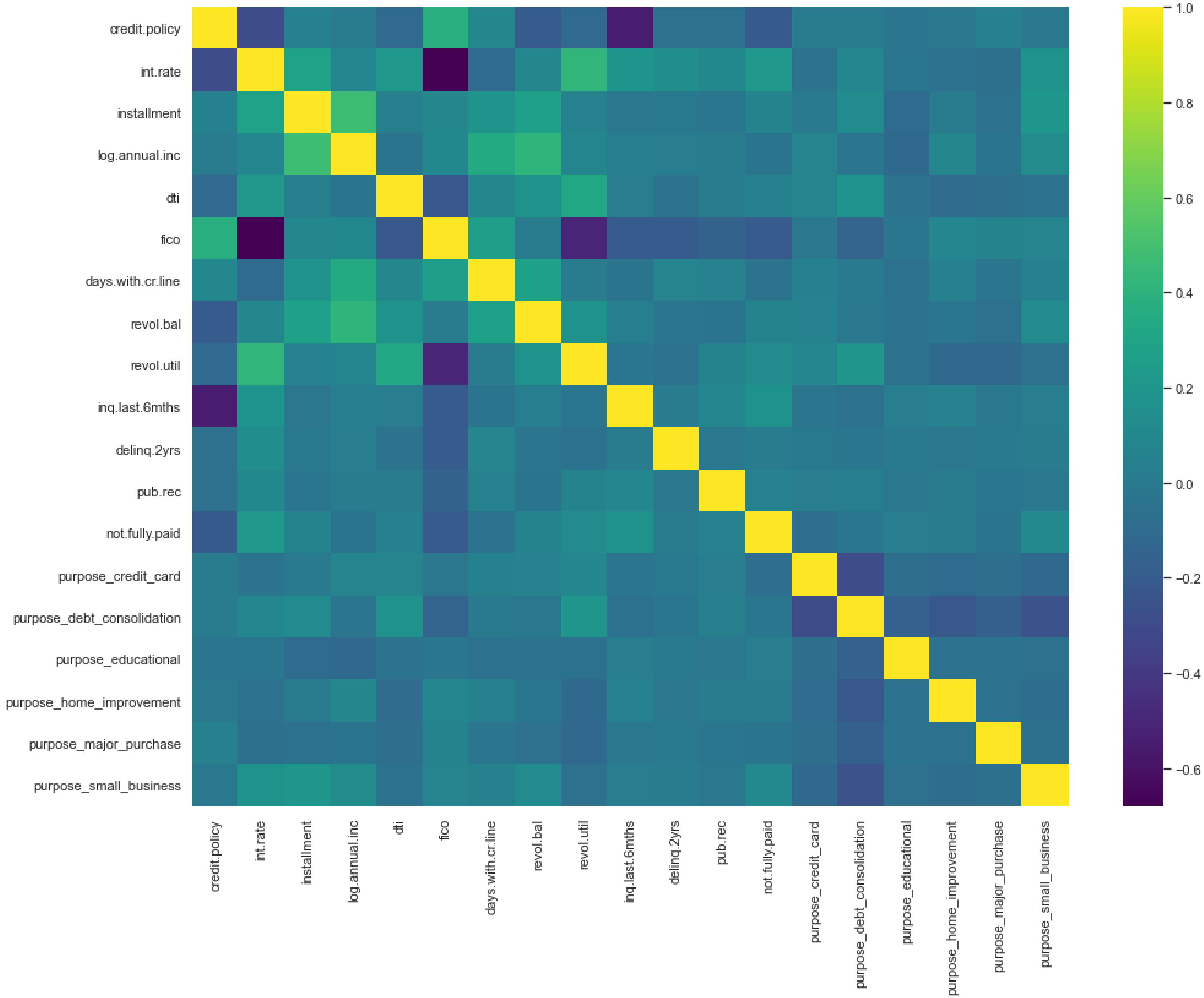
```
dti                  0.337109          0.029189        0.021792  0.006209
fico                 0.541289          0.185293        0.216340  0.147592
days.with.cr.line    0.024239          0.041736        0.081374  0.071826
revol.bal            0.203779          0.022394        0.033243  0.031010
revol.util                NaN          0.013880        0.042740  0.066717
inq.last.6mths            NaN               NaN        0.021245  0.072673
delinq.2yrs               NaN               NaN             NaN  0.009184
pub.rec                   NaN               NaN             NaN       NaN
not.fully.paid            NaN               NaN             NaN       NaN

                    not.fully.paid
credit.policy             0.158119
int.rate                  0.159552
installment               0.049955
log.annual.inc            0.033439
dti                       0.037362
fico                      0.149666
days.with.cr.line         0.029237
revol.bal                 0.053699
revol.util                0.082088
inq.last.6mths            0.149452
delinq.2yrs               0.008881
pub.rec                   0.048634
not.fully.paid                 NaN
```

In [95]:

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

Out[95]:

```
<AxesSubplot:>
```



In [62]:

```
loan.describe().transpose()
```

Out[62]:

	count	mean	std	min	25%	50%	75%	max
credit.policy	9578.0	0.804970	0.396245	0.000000	1.000000	1.000000	1.000000	1.000000e+00
int.rate	9578.0	0.122640	0.026847	0.060000	0.103900	0.122100	0.140700	2.164000e-01
installment	9578.0	319.089413	207.071301	15.670000	163.770000	268.950000	432.762500	9.401400e+02
log.annual.inc	9578.0	10.932117	0.614813	7.547502	10.558414	10.928884	11.291293	1.452835e+01
dti	9578.0	12.606679	6.883970	0.000000	7.212500	12.665000	17.950000	2.996000e+01
fico	9578.0	710.846314	37.970537	612.000000	682.000000	707.000000	737.000000	8.270000e+02
days.with.cr.line	9578.0	4560.767197	2496.930377	178.958333	2820.000000	4139.958333	5730.000000	1.763996e+04
revol.bal	9578.0	16913.963876	33756.189557	0.000000	3187.000000	8596.000000	18249.500000	1.207359e+06

revol.util	9578.0	46.799236	29.014417	0.000000	22.600000	46.300000	70.900000	1.190000e+02
inq.last.6mths	9578.0	1.577469	2.200245	0.000000	0.000000	1.000000	2.000000	3.300000e+01
delinq.2yrs	9578.0	0.163708	0.546215	0.000000	0.000000	0.000000	0.000000	1.300000e+01
pub.rec	9578.0	0.062122	0.262126	0.000000	0.000000	0.000000	0.000000	5.000000e+00
not.fully.paid	9578.0	0.160054	0.366676	0.000000	0.000000	0.000000	0.000000	1.000000e+00

In [63]:

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

Out[63]:

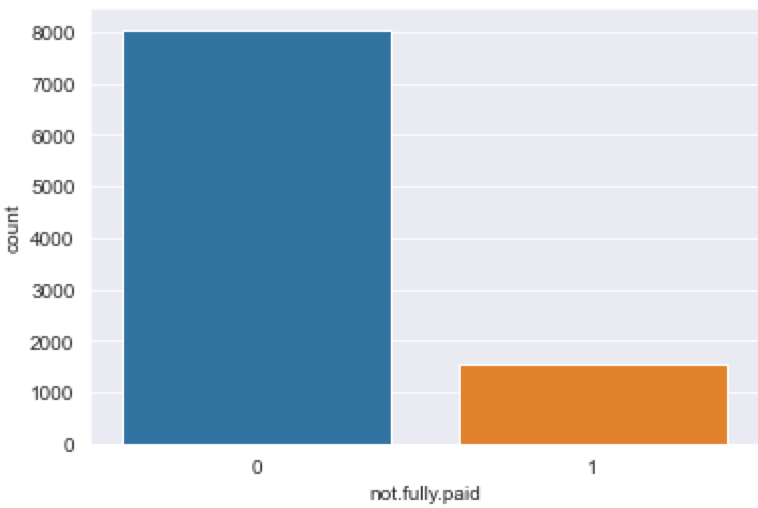
```
not.fully.paid
0    0.839946
1    0.160054
Name: not.fully.paid, dtype: float64
```

In [64]:

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

Out[64]:

```
<AxesSubplot:xlabel='not.fully.paid', ylabel='count'>
```



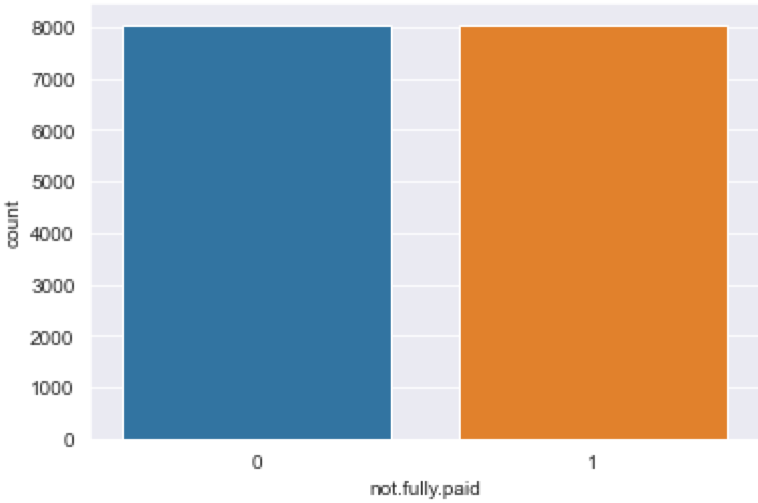
In [65]:

```
count_class_0, count_class_1 = loan['not.fully.paid'].value_counts()
loan_0 = loan[loan['not.fully.paid'] == 0]
loan_1 = loan[loan['not.fully.paid'] == 1]
loan_1_over = loan_1.sample(count_class_0, replace=True)
loan_test_over = pd.concat([loan_0, loan_1_over], axis=0)
print('Random over-sampling:')
print(loan_test_over['not.fully.paid'].value_counts())

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

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

Out[65]:

```
<AxesSubplot:xlabel='not.fully.paid', ylabel='count'>
```



In [66]:

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16090 entries, 0 to 1885
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   credit.policy               16090 non-null  int64  
 1   int.rate                    16090 non-null  float64
 2   installment                 16090 non-null  float64
 3   log.annual.inc              16090 non-null  float64
 4   dti                         16090 non-null  float64
 5   fico                        16090 non-null  int64  
 6   days.with.cr.line           16090 non-null  float64
 7   revol.bal                   16090 non-null  int64  
 8   revol.util                  16090 non-null  float64
 9   inq.last.6mths              16090 non-null  int64  
 10  delinq.2yrs                 16090 non-null  int64  
 11  pub.rec                     16090 non-null  int64  
 12  not.fully.paid              16090 non-null  int64  
 13  purpose_credit_card         16090 non-null  uint8  
 14  purpose_debt_consolidation  16090 non-null  uint8  
 15  purpose_educational         16090 non-null  uint8  
 16  purpose_home_improvement    16090 non-null  uint8  
 17  purpose_major_purchase      16090 non-null  uint8  
 18  purpose_small_business      16090 non-null  uint8  
dtypes: float64(6), int64(7), uint8(6)
memory usage: 2.1 MB
```

In [67]:

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

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

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

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

model = Sequential()
```

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

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

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


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

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

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

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

```
Epoch 1/200
44/44 [==============================] - 0s 3ms/step - loss: 0.6905 - accuracy: 0.5214 - val
_loss: 0.6860 - val_accuracy: 0.5728
Epoch 2/200
44/44 [==============================] - 0s 772us/step - loss: 0.6822 - accuracy: 0.5833 - v
al_loss: 0.6788 - val_accuracy: 0.6008
Epoch 3/200
44/44 [==============================] - 0s 772us/step - loss: 0.6753 - accuracy: 0.6002 - v
al_loss: 0.6728 - val_accuracy: 0.6068
Epoch 4/200
44/44 [==============================] - 0s 795us/step - loss: 0.6685 - accuracy: 0.6066 - v
al_loss: 0.6679 - val_accuracy: 0.6089
Epoch 5/200
44/44 [==============================] - 0s 795us/step - loss: 0.6614 - accuracy: 0.6145 - v
al_loss: 0.6593 - val_accuracy: 0.6018
Epoch 6/200
44/44 [==============================] - 0s 795us/step - loss: 0.6501 - accuracy: 0.6169 - v
al_loss: 0.6531 - val_accuracy: 0.6027
Epoch 7/200
44/44 [==============================] - 0s 795us/step - loss: 0.6444 - accuracy: 0.6188 - v
al_loss: 0.6506 - val_accuracy: 0.6116
```

```
Epoch 8/200
44/44 [==============================] - 0s 795us/step - loss: 0.6413 - accuracy: 0.6213 - v
al_loss: 0.6497 - val_accuracy: 0.6091
Epoch 9/200
44/44 [==============================] - 0s 772us/step - loss: 0.6400 - accuracy: 0.6239 - v
al_loss: 0.6485 - val_accuracy: 0.6087
Epoch 10/200
44/44 [==============================] - 0s 772us/step - loss: 0.6382 - accuracy: 0.6267 - v
al_loss: 0.6473 - val_accuracy: 0.6124
Epoch 11/200
44/44 [==============================] - 0s 795us/step - loss: 0.6370 - accuracy: 0.6276 - v
al_loss: 0.6475 - val_accuracy: 0.6060
Epoch 12/200
44/44 [==============================] - 0s 772us/step - loss: 0.6355 - accuracy: 0.6287 - v
al_loss: 0.6460 - val_accuracy: 0.6167
Epoch 13/200
44/44 [==============================] - 0s 795us/step - loss: 0.6341 - accuracy: 0.6310 - v
al_loss: 0.6470 - val_accuracy: 0.6157
Epoch 14/200
44/44 [==============================] - 0s 772us/step - loss: 0.6338 - accuracy: 0.6314 - v
al_loss: 0.6450 - val_accuracy: 0.6153
Epoch 15/200
44/44 [==============================] - 0s 795us/step - loss: 0.6323 - accuracy: 0.6322 - v
al_loss: 0.6450 - val_accuracy: 0.6122
Epoch 16/200
44/44 [==============================] - 0s 772us/step - loss: 0.6314 - accuracy: 0.6320 - v
al_loss: 0.6441 - val_accuracy: 0.6169
Epoch 17/200
44/44 [==============================] - 0s 771us/step - loss: 0.6310 - accuracy: 0.6332 - v
al_loss: 0.6434 - val_accuracy: 0.6182
Epoch 18/200
44/44 [==============================] - 0s 772us/step - loss: 0.6306 - accuracy: 0.6341 - v
al_loss: 0.6431 - val_accuracy: 0.6201
Epoch 19/200
44/44 [==============================] - 0s 772us/step - loss: 0.6310 - accuracy: 0.6322 - v
al_loss: 0.6434 - val_accuracy: 0.6240
Epoch 20/200
44/44 [==============================] - 0s 772us/step - loss: 0.6303 - accuracy: 0.6352 - v
al_loss: 0.6427 - val_accuracy: 0.6159
Epoch 21/200
44/44 [==============================] - 0s 795us/step - loss: 0.6285 - accuracy: 0.6354 - v
al_loss: 0.6425 - val_accuracy: 0.6143
Epoch 22/200
44/44 [==============================] - 0s 795us/step - loss: 0.6278 - accuracy: 0.6354 - v
al_loss: 0.6424 - val_accuracy: 0.6198
Epoch 23/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6272 - accuracy: 0.6376 - val
_loss: 0.6414 - val_accuracy: 0.6176
Epoch 24/200
44/44 [==============================] - 0s 999us/step - loss: 0.6267 - accuracy: 0.6364 - v
al_loss: 0.6420 - val_accuracy: 0.6178
Epoch 25/200
44/44 [==============================] - 0s 863us/step - loss: 0.6265 - accuracy: 0.6366 - v
al_loss: 0.6407 - val_accuracy: 0.6238
Epoch 26/200
44/44 [==============================] - 0s 818us/step - loss: 0.6259 - accuracy: 0.6376 - v
al_loss: 0.6415 - val_accuracy: 0.6196
Epoch 27/200
44/44 [==============================] - 0s 863us/step - loss: 0.6254 - accuracy: 0.6388 - v
al_loss: 0.6400 - val_accuracy: 0.6252
Epoch 28/200
44/44 [==============================] - 0s 772us/step - loss: 0.6248 - accuracy: 0.6404 - v
al_loss: 0.6395 - val_accuracy: 0.6306
Epoch 29/200
44/44 [==============================] - 0s 772us/step - loss: 0.6250 - accuracy: 0.6419 - v
al_loss: 0.6394 - val_accuracy: 0.6225
Epoch 30/200
44/44 [==============================] - 0s 772us/step - loss: 0.6238 - accuracy: 0.6425 - v
al_loss: 0.6394 - val_accuracy: 0.6238
```

```
Epoch 31/200
44/44 [==============================] - 0s 795us/step - loss: 0.6235 - accuracy: 0.6417 - v
al_loss: 0.6383 - val_accuracy: 0.6263
Epoch 32/200
44/44 [==============================] - 0s 772us/step - loss: 0.6227 - accuracy: 0.6454 - v
al_loss: 0.6398 - val_accuracy: 0.6292
Epoch 33/200
44/44 [==============================] - 0s 772us/step - loss: 0.6223 - accuracy: 0.6436 - v
al_loss: 0.6386 - val_accuracy: 0.6290
Epoch 34/200
44/44 [==============================] - 0s 818us/step - loss: 0.6215 - accuracy: 0.6476 - v
al_loss: 0.6381 - val_accuracy: 0.6259
Epoch 35/200
44/44 [==============================] - 0s 795us/step - loss: 0.6209 - accuracy: 0.6457 - v
al_loss: 0.6391 - val_accuracy: 0.6267
Epoch 36/200
44/44 [==============================] - 0s 772us/step - loss: 0.6215 - accuracy: 0.6427 - v
al_loss: 0.6369 - val_accuracy: 0.6290
Epoch 37/200
44/44 [==============================] - 0s 795us/step - loss: 0.6206 - accuracy: 0.6462 - v
al_loss: 0.6365 - val_accuracy: 0.6321
Epoch 38/200
44/44 [==============================] - 0s 772us/step - loss: 0.6208 - accuracy: 0.6497 - v
al_loss: 0.6374 - val_accuracy: 0.6321
Epoch 39/200
44/44 [==============================] - 0s 772us/step - loss: 0.6198 - accuracy: 0.6507 - v
al_loss: 0.6374 - val_accuracy: 0.6317
Epoch 40/200
44/44 [==============================] - 0s 818us/step - loss: 0.6192 - accuracy: 0.6492 - v
al_loss: 0.6359 - val_accuracy: 0.6329
Epoch 41/200
44/44 [==============================] - 0s 818us/step - loss: 0.6186 - accuracy: 0.6488 - v
al_loss: 0.6357 - val_accuracy: 0.6348
Epoch 42/200
44/44 [==============================] - 0s 795us/step - loss: 0.6182 - accuracy: 0.6512 - v
al_loss: 0.6371 - val_accuracy: 0.6312
Epoch 43/200
44/44 [==============================] - 0s 818us/step - loss: 0.6179 - accuracy: 0.6530 - v
al_loss: 0.6361 - val_accuracy: 0.6352
Epoch 44/200
44/44 [==============================] - 0s 795us/step - loss: 0.6175 - accuracy: 0.6529 - v
al_loss: 0.6358 - val_accuracy: 0.6352
Epoch 45/200
44/44 [==============================] - 0s 772us/step - loss: 0.6177 - accuracy: 0.6536 - v
al_loss: 0.6357 - val_accuracy: 0.6350
Epoch 46/200
44/44 [==============================] - 0s 795us/step - loss: 0.6168 - accuracy: 0.6536 - v
al_loss: 0.6365 - val_accuracy: 0.6321
Epoch 47/200
44/44 [==============================] - 0s 795us/step - loss: 0.6168 - accuracy: 0.6543 - v
al_loss: 0.6353 - val_accuracy: 0.6377
Epoch 48/200
44/44 [==============================] - 0s 772us/step - loss: 0.6155 - accuracy: 0.6543 - v
al_loss: 0.6343 - val_accuracy: 0.6399
Epoch 49/200
44/44 [==============================] - 0s 795us/step - loss: 0.6154 - accuracy: 0.6552 - v
al_loss: 0.6339 - val_accuracy: 0.6414
Epoch 50/200
44/44 [==============================] - 0s 795us/step - loss: 0.6148 - accuracy: 0.6579 - v
al_loss: 0.6340 - val_accuracy: 0.6381
Epoch 51/200
44/44 [==============================] - 0s 818us/step - loss: 0.6147 - accuracy: 0.6566 - v
al_loss: 0.6338 - val_accuracy: 0.6385
Epoch 52/200
44/44 [==============================] - 0s 772us/step - loss: 0.6145 - accuracy: 0.6582 - v
al_loss: 0.6344 - val_accuracy: 0.6414
Epoch 53/200
44/44 [==============================] - 0s 795us/step - loss: 0.6146 - accuracy: 0.6568 - v
al_loss: 0.6335 - val_accuracy: 0.6393
```

```
Epoch 54/200
44/44 [==============================] - 0s 795us/step - loss: 0.6132 - accuracy: 0.6576 - v
al_loss: 0.6345 - val_accuracy: 0.6414
Epoch 55/200
44/44 [==============================] - 0s 818us/step - loss: 0.6133 - accuracy: 0.6550 - v
al_loss: 0.6339 - val_accuracy: 0.6453
Epoch 56/200
44/44 [==============================] - 0s 772us/step - loss: 0.6140 - accuracy: 0.6569 - v
al_loss: 0.6375 - val_accuracy: 0.6275
Epoch 57/200
44/44 [==============================] - 0s 795us/step - loss: 0.6131 - accuracy: 0.6545 - v
al_loss: 0.6334 - val_accuracy: 0.6401
Epoch 58/200
44/44 [==============================] - 0s 772us/step - loss: 0.6128 - accuracy: 0.6582 - v
al_loss: 0.6331 - val_accuracy: 0.6453
Epoch 59/200
44/44 [==============================] - 0s 773us/step - loss: 0.6123 - accuracy: 0.6573 - v
al_loss: 0.6336 - val_accuracy: 0.6404
Epoch 60/200
44/44 [==============================] - 0s 795us/step - loss: 0.6120 - accuracy: 0.6587 - v
al_loss: 0.6336 - val_accuracy: 0.6459
Epoch 61/200
44/44 [==============================] - 0s 772us/step - loss: 0.6124 - accuracy: 0.6595 - v
al_loss: 0.6326 - val_accuracy: 0.6443
Epoch 62/200
44/44 [==============================] - 0s 772us/step - loss: 0.6112 - accuracy: 0.6606 - v
al_loss: 0.6333 - val_accuracy: 0.6420
Epoch 63/200
44/44 [==============================] - 0s 818us/step - loss: 0.6108 - accuracy: 0.6595 - v
al_loss: 0.6332 - val_accuracy: 0.6443
Epoch 64/200
44/44 [==============================] - 0s 772us/step - loss: 0.6109 - accuracy: 0.6628 - v
al_loss: 0.6320 - val_accuracy: 0.6478
Epoch 65/200
44/44 [==============================] - 0s 772us/step - loss: 0.6106 - accuracy: 0.6624 - v
al_loss: 0.6333 - val_accuracy: 0.6372
Epoch 66/200
44/44 [==============================] - 0s 863us/step - loss: 0.6103 - accuracy: 0.6599 - v
al_loss: 0.6322 - val_accuracy: 0.6459
Epoch 67/200
44/44 [==============================] - 0s 840us/step - loss: 0.6112 - accuracy: 0.6600 - v
al_loss: 0.6318 - val_accuracy: 0.6459
Epoch 68/200
44/44 [==============================] - 0s 772us/step - loss: 0.6103 - accuracy: 0.6608 - v
al_loss: 0.6319 - val_accuracy: 0.6466
Epoch 69/200
44/44 [==============================] - 0s 772us/step - loss: 0.6108 - accuracy: 0.6585 - v
al_loss: 0.6333 - val_accuracy: 0.6491
Epoch 70/200
44/44 [==============================] - 0s 772us/step - loss: 0.6098 - accuracy: 0.6591 - v
al_loss: 0.6324 - val_accuracy: 0.6464
Epoch 71/200
44/44 [==============================] - 0s 772us/step - loss: 0.6092 - accuracy: 0.6595 - v
al_loss: 0.6338 - val_accuracy: 0.6358
Epoch 72/200
44/44 [==============================] - 0s 795us/step - loss: 0.6099 - accuracy: 0.6584 - v
al_loss: 0.6315 - val_accuracy: 0.6424
Epoch 73/200
44/44 [==============================] - 0s 772us/step - loss: 0.6089 - accuracy: 0.6598 - v
al_loss: 0.6320 - val_accuracy: 0.6499
Epoch 74/200
44/44 [==============================] - 0s 794us/step - loss: 0.6086 - accuracy: 0.6613 - v
al_loss: 0.6331 - val_accuracy: 0.6466
Epoch 75/200
44/44 [==============================] - 0s 772us/step - loss: 0.6089 - accuracy: 0.6631 - v
al_loss: 0.6311 - val_accuracy: 0.6443
Epoch 76/200
44/44 [==============================] - 0s 772us/step - loss: 0.6082 - accuracy: 0.6622 - v
al_loss: 0.6316 - val_accuracy: 0.6418
```

```
Epoch 77/200
44/44 [==============================] - 0s 772us/step - loss: 0.6079 - accuracy: 0.6586 - v
al_loss: 0.6304 - val_accuracy: 0.6453
Epoch 78/200
44/44 [==============================] - 0s 772us/step - loss: 0.6075 - accuracy: 0.6604 - v
al_loss: 0.6306 - val_accuracy: 0.6507
Epoch 79/200
44/44 [==============================] - 0s 772us/step - loss: 0.6075 - accuracy: 0.6592 - v
al_loss: 0.6313 - val_accuracy: 0.6377
Epoch 80/200
44/44 [==============================] - 0s 772us/step - loss: 0.6071 - accuracy: 0.6602 - v
al_loss: 0.6316 - val_accuracy: 0.6385
Epoch 81/200
44/44 [==============================] - 0s 795us/step - loss: 0.6071 - accuracy: 0.6608 - v
al_loss: 0.6301 - val_accuracy: 0.6455
Epoch 82/200
44/44 [==============================] - 0s 818us/step - loss: 0.6069 - accuracy: 0.6647 - v
al_loss: 0.6303 - val_accuracy: 0.6416
Epoch 83/200
44/44 [==============================] - 0s 795us/step - loss: 0.6068 - accuracy: 0.6623 - v
al_loss: 0.6301 - val_accuracy: 0.6489
Epoch 84/200
44/44 [==============================] - 0s 750us/step - loss: 0.6062 - accuracy: 0.6635 - v
al_loss: 0.6301 - val_accuracy: 0.6520
Epoch 85/200
44/44 [==============================] - 0s 772us/step - loss: 0.6061 - accuracy: 0.6641 - v
al_loss: 0.6302 - val_accuracy: 0.6439
Epoch 86/200
44/44 [==============================] - 0s 772us/step - loss: 0.6057 - accuracy: 0.6631 - v
al_loss: 0.6307 - val_accuracy: 0.6509
Epoch 87/200
44/44 [==============================] - 0s 772us/step - loss: 0.6068 - accuracy: 0.6585 - v
al_loss: 0.6314 - val_accuracy: 0.6499
Epoch 88/200
44/44 [==============================] - 0s 795us/step - loss: 0.6058 - accuracy: 0.6638 - v
al_loss: 0.6294 - val_accuracy: 0.6495
Epoch 89/200
44/44 [==============================] - 0s 772us/step - loss: 0.6049 - accuracy: 0.6650 - v
al_loss: 0.6297 - val_accuracy: 0.6520
Epoch 90/200
44/44 [==============================] - 0s 772us/step - loss: 0.6054 - accuracy: 0.6654 - v
al_loss: 0.6294 - val_accuracy: 0.6484
Epoch 91/200
44/44 [==============================] - 0s 818us/step - loss: 0.6055 - accuracy: 0.6630 - v
al_loss: 0.6303 - val_accuracy: 0.6499
Epoch 92/200
44/44 [==============================] - 0s 772us/step - loss: 0.6042 - accuracy: 0.6671 - v
al_loss: 0.6297 - val_accuracy: 0.6491
Epoch 93/200
44/44 [==============================] - 0s 818us/step - loss: 0.6044 - accuracy: 0.6650 - v
al_loss: 0.6294 - val_accuracy: 0.6534
Epoch 94/200
44/44 [==============================] - 0s 818us/step - loss: 0.6038 - accuracy: 0.6647 - v
al_loss: 0.6297 - val_accuracy: 0.6466
Epoch 95/200
44/44 [==============================] - 0s 818us/step - loss: 0.6046 - accuracy: 0.6668 - v
al_loss: 0.6303 - val_accuracy: 0.6426
Epoch 96/200
44/44 [==============================] - 0s 817us/step - loss: 0.6035 - accuracy: 0.6654 - v
al_loss: 0.6285 - val_accuracy: 0.6540
Epoch 97/200
44/44 [==============================] - 0s 795us/step - loss: 0.6037 - accuracy: 0.6679 - v
al_loss: 0.6297 - val_accuracy: 0.6435
Epoch 98/200
44/44 [==============================] - 0s 795us/step - loss: 0.6037 - accuracy: 0.6655 - v
al_loss: 0.6281 - val_accuracy: 0.6472
Epoch 99/200
44/44 [==============================] - 0s 795us/step - loss: 0.6034 - accuracy: 0.6663 - v
al_loss: 0.6290 - val_accuracy: 0.6420
```

```
Epoch 100/200
44/44 [==============================] - 0s 772us/step - loss: 0.6033 - accuracy: 0.6682 - v
al_loss: 0.6298 - val_accuracy: 0.6401
Epoch 101/200
44/44 [==============================] - 0s 749us/step - loss: 0.6026 - accuracy: 0.6639 - v
al_loss: 0.6272 - val_accuracy: 0.6464
Epoch 102/200
44/44 [==============================] - 0s 795us/step - loss: 0.6021 - accuracy: 0.6675 - v
al_loss: 0.6268 - val_accuracy: 0.6536
Epoch 103/200
44/44 [==============================] - 0s 772us/step - loss: 0.6026 - accuracy: 0.6657 - v
al_loss: 0.6272 - val_accuracy: 0.6466
Epoch 104/200
44/44 [==============================] - 0s 749us/step - loss: 0.6014 - accuracy: 0.6675 - v
al_loss: 0.6273 - val_accuracy: 0.6424
Epoch 105/200
44/44 [==============================] - 0s 772us/step - loss: 0.6013 - accuracy: 0.6679 - v
al_loss: 0.6272 - val_accuracy: 0.6451
Epoch 106/200
44/44 [==============================] - 0s 772us/step - loss: 0.6016 - accuracy: 0.6659 - v
al_loss: 0.6289 - val_accuracy: 0.6399
Epoch 107/200
44/44 [==============================] - 0s 772us/step - loss: 0.6012 - accuracy: 0.6700 - v
al_loss: 0.6272 - val_accuracy: 0.6486
Epoch 108/200
44/44 [==============================] - 0s 772us/step - loss: 0.6011 - accuracy: 0.6642 - v
al_loss: 0.6263 - val_accuracy: 0.6476
Epoch 109/200
44/44 [==============================] - 0s 772us/step - loss: 0.6007 - accuracy: 0.6699 - v
al_loss: 0.6267 - val_accuracy: 0.6567
Epoch 110/200
44/44 [==============================] - 0s 772us/step - loss: 0.6008 - accuracy: 0.6660 - v
al_loss: 0.6276 - val_accuracy: 0.6526
Epoch 111/200
44/44 [==============================] - 0s 772us/step - loss: 0.6001 - accuracy: 0.6676 - v
al_loss: 0.6269 - val_accuracy: 0.6513
Epoch 112/200
44/44 [==============================] - ETA: 0s - loss: 0.6210 - accuracy: 0.65 - 0s 772us/
step - loss: 0.5997 - accuracy: 0.6686 - val_loss: 0.6257 - val_accuracy: 0.6532
Epoch 113/200
44/44 [==============================] - 0s 840us/step - loss: 0.6001 - accuracy: 0.6672 - v
al_loss: 0.6264 - val_accuracy: 0.6578
Epoch 114/200
44/44 [==============================] - 0s 818us/step - loss: 0.5995 - accuracy: 0.6686 - v
al_loss: 0.6257 - val_accuracy: 0.6495
Epoch 115/200
44/44 [==============================] - 0s 818us/step - loss: 0.6006 - accuracy: 0.6707 - v
al_loss: 0.6270 - val_accuracy: 0.6401
Epoch 116/200
44/44 [==============================] - 0s 818us/step - loss: 0.6001 - accuracy: 0.6701 - v
al_loss: 0.6265 - val_accuracy: 0.6536
Epoch 117/200
44/44 [==============================] - 0s 818us/step - loss: 0.5982 - accuracy: 0.6714 - v
al_loss: 0.6260 - val_accuracy: 0.6478
Epoch 118/200
44/44 [==============================] - 0s 795us/step - loss: 0.5985 - accuracy: 0.6682 - v
al_loss: 0.6258 - val_accuracy: 0.6571
Epoch 119/200
44/44 [==============================] - 0s 863us/step - loss: 0.5996 - accuracy: 0.6700 - v
al_loss: 0.6243 - val_accuracy: 0.6522
Epoch 120/200
44/44 [==============================] - 0s 795us/step - loss: 0.5981 - accuracy: 0.6679 - v
al_loss: 0.6269 - val_accuracy: 0.6420
Epoch 121/200
44/44 [==============================] - 0s 863us/step - loss: 0.5987 - accuracy: 0.6702 - v
al_loss: 0.6261 - val_accuracy: 0.6534
Epoch 122/200
44/44 [==============================] - 0s 795us/step - loss: 0.5983 - accuracy: 0.6707 - v
al_loss: 0.6255 - val_accuracy: 0.6470
```

```
Epoch 123/200
44/44 [==============================] - 0s 795us/step - loss: 0.5976 - accuracy: 0.6697 - v
al_loss: 0.6253 - val_accuracy: 0.6534
Epoch 124/200
44/44 [==============================] - 0s 772us/step - loss: 0.5976 - accuracy: 0.6685 - v
al_loss: 0.6256 - val_accuracy: 0.6555
Epoch 125/200
44/44 [==============================] - 0s 795us/step - loss: 0.5972 - accuracy: 0.6702 - v
al_loss: 0.6249 - val_accuracy: 0.6530
Epoch 126/200
44/44 [==============================] - 0s 795us/step - loss: 0.5967 - accuracy: 0.6724 - v
al_loss: 0.6276 - val_accuracy: 0.6567
Epoch 127/200
44/44 [==============================] - 0s 795us/step - loss: 0.5972 - accuracy: 0.6710 - v
al_loss: 0.6269 - val_accuracy: 0.6457
Epoch 128/200
44/44 [==============================] - 0s 772us/step - loss: 0.5966 - accuracy: 0.6722 - v
al_loss: 0.6245 - val_accuracy: 0.6544
Epoch 129/200
44/44 [==============================] - 0s 772us/step - loss: 0.5958 - accuracy: 0.6693 - v
al_loss: 0.6243 - val_accuracy: 0.6522
Epoch 130/200
44/44 [==============================] - 0s 818us/step - loss: 0.5963 - accuracy: 0.6719 - v
al_loss: 0.6254 - val_accuracy: 0.6507
Epoch 131/200
44/44 [==============================] - 0s 863us/step - loss: 0.5964 - accuracy: 0.6681 - v
al_loss: 0.6263 - val_accuracy: 0.6540
Epoch 132/200
44/44 [==============================] - 0s 795us/step - loss: 0.5967 - accuracy: 0.6694 - v
al_loss: 0.6241 - val_accuracy: 0.6578
Epoch 133/200
44/44 [==============================] - 0s 795us/step - loss: 0.5956 - accuracy: 0.6710 - v
al_loss: 0.6241 - val_accuracy: 0.6536
Epoch 134/200
44/44 [==============================] - 0s 795us/step - loss: 0.5959 - accuracy: 0.6707 - v
al_loss: 0.6241 - val_accuracy: 0.6470
Epoch 135/200
44/44 [==============================] - 0s 795us/step - loss: 0.5957 - accuracy: 0.6691 - v
al_loss: 0.6240 - val_accuracy: 0.6505
Epoch 136/200
44/44 [==============================] - 0s 818us/step - loss: 0.5966 - accuracy: 0.6673 - v
al_loss: 0.6263 - val_accuracy: 0.6559
Epoch 137/200
44/44 [==============================] - 0s 795us/step - loss: 0.5956 - accuracy: 0.6711 - v
al_loss: 0.6248 - val_accuracy: 0.6559
Epoch 138/200
44/44 [==============================] - 0s 772us/step - loss: 0.5955 - accuracy: 0.6705 - v
al_loss: 0.6237 - val_accuracy: 0.6480
Epoch 139/200
44/44 [==============================] - 0s 772us/step - loss: 0.5948 - accuracy: 0.6726 - v
al_loss: 0.6260 - val_accuracy: 0.6424
Epoch 140/200
44/44 [==============================] - 0s 773us/step - loss: 0.5942 - accuracy: 0.6718 - v
al_loss: 0.6252 - val_accuracy: 0.6468
Epoch 141/200
44/44 [==============================] - 0s 772us/step - loss: 0.5950 - accuracy: 0.6723 - v
al_loss: 0.6242 - val_accuracy: 0.6497
Epoch 142/200
44/44 [==============================] - 0s 772us/step - loss: 0.5939 - accuracy: 0.6712 - v
al_loss: 0.6242 - val_accuracy: 0.6457
Epoch 143/200
44/44 [==============================] - 0s 795us/step - loss: 0.5941 - accuracy: 0.6725 - v
al_loss: 0.6242 - val_accuracy: 0.6528
Epoch 144/200
44/44 [==============================] - 0s 749us/step - loss: 0.5933 - accuracy: 0.6735 - v
al_loss: 0.6240 - val_accuracy: 0.6497
Epoch 145/200
44/44 [==============================] - 0s 772us/step - loss: 0.5939 - accuracy: 0.6735 - v
al_loss: 0.6260 - val_accuracy: 0.6428
```

```
Epoch 146/200
44/44 [==============================] - 0s 772us/step - loss: 0.5928 - accuracy: 0.6741 - v
al_loss: 0.6249 - val_accuracy: 0.6509
Epoch 147/200
44/44 [==============================] - 0s 772us/step - loss: 0.5929 - accuracy: 0.6715 - v
al_loss: 0.6252 - val_accuracy: 0.6453
Epoch 148/200
44/44 [==============================] - 0s 795us/step - loss: 0.5939 - accuracy: 0.6697 - v
al_loss: 0.6249 - val_accuracy: 0.6482
Epoch 149/200
44/44 [==============================] - 0s 773us/step - loss: 0.5924 - accuracy: 0.6724 - v
al_loss: 0.6223 - val_accuracy: 0.6524
Epoch 150/200
44/44 [==============================] - 0s 771us/step - loss: 0.5924 - accuracy: 0.6710 - v
al_loss: 0.6234 - val_accuracy: 0.6470
Epoch 151/200
44/44 [==============================] - 0s 795us/step - loss: 0.5921 - accuracy: 0.6731 - v
al_loss: 0.6237 - val_accuracy: 0.6449
Epoch 152/200
44/44 [==============================] - 0s 772us/step - loss: 0.5924 - accuracy: 0.6746 - v
al_loss: 0.6233 - val_accuracy: 0.6497
Epoch 153/200
44/44 [==============================] - 0s 772us/step - loss: 0.5918 - accuracy: 0.6730 - v
al_loss: 0.6229 - val_accuracy: 0.6526
Epoch 154/200
44/44 [==============================] - 0s 772us/step - loss: 0.5912 - accuracy: 0.6748 - v
al_loss: 0.6225 - val_accuracy: 0.6507
Epoch 155/200
44/44 [==============================] - 0s 795us/step - loss: 0.5925 - accuracy: 0.6744 - v
al_loss: 0.6265 - val_accuracy: 0.6433
Epoch 156/200
44/44 [==============================] - 0s 795us/step - loss: 0.5930 - accuracy: 0.6738 - v
al_loss: 0.6264 - val_accuracy: 0.6447
Epoch 157/200
44/44 [==============================] - 0s 818us/step - loss: 0.5913 - accuracy: 0.6755 - v
al_loss: 0.6229 - val_accuracy: 0.6505
Epoch 158/200
44/44 [==============================] - 0s 795us/step - loss: 0.5907 - accuracy: 0.6747 - v
al_loss: 0.6231 - val_accuracy: 0.6459
Epoch 159/200
44/44 [==============================] - 0s 772us/step - loss: 0.5906 - accuracy: 0.6720 - v
al_loss: 0.6237 - val_accuracy: 0.6547
Epoch 160/200
44/44 [==============================] - 0s 795us/step - loss: 0.5904 - accuracy: 0.6731 - v
al_loss: 0.6235 - val_accuracy: 0.6526
Epoch 161/200
44/44 [==============================] - 0s 772us/step - loss: 0.5903 - accuracy: 0.6758 - v
al_loss: 0.6238 - val_accuracy: 0.6470
Epoch 162/200
44/44 [==============================] - 0s 795us/step - loss: 0.5900 - accuracy: 0.6769 - v
al_loss: 0.6224 - val_accuracy: 0.6530
Epoch 163/200
44/44 [==============================] - 0s 773us/step - loss: 0.5894 - accuracy: 0.6739 - v
al_loss: 0.6233 - val_accuracy: 0.6470
Epoch 164/200
44/44 [==============================] - 0s 749us/step - loss: 0.5905 - accuracy: 0.6747 - v
al_loss: 0.6229 - val_accuracy: 0.6449
Epoch 165/200
44/44 [==============================] - 0s 795us/step - loss: 0.5899 - accuracy: 0.6750 - v
al_loss: 0.6231 - val_accuracy: 0.6482
Epoch 166/200
44/44 [==============================] - 0s 840us/step - loss: 0.5899 - accuracy: 0.6764 - v
al_loss: 0.6226 - val_accuracy: 0.6478
Epoch 167/200
44/44 [==============================] - 0s 795us/step - loss: 0.5905 - accuracy: 0.6751 - v
al_loss: 0.6217 - val_accuracy: 0.6499
Epoch 168/200
44/44 [==============================] - 0s 818us/step - loss: 0.5895 - accuracy: 0.6753 - v
al_loss: 0.6223 - val_accuracy: 0.6544
```

```
Epoch 169/200
44/44 [==============================] - 0s 795us/step - loss: 0.5899 - accuracy: 0.6761 - v
al_loss: 0.6215 - val_accuracy: 0.6559
Epoch 170/200
44/44 [==============================] - 0s 795us/step - loss: 0.5881 - accuracy: 0.6797 - v
al_loss: 0.6206 - val_accuracy: 0.6482
Epoch 171/200
44/44 [==============================] - 0s 772us/step - loss: 0.5887 - accuracy: 0.6766 - v
al_loss: 0.6214 - val_accuracy: 0.6518
Epoch 172/200
44/44 [==============================] - 0s 772us/step - loss: 0.5887 - accuracy: 0.6765 - v
al_loss: 0.6228 - val_accuracy: 0.6464
Epoch 173/200
44/44 [==============================] - 0s 772us/step - loss: 0.5886 - accuracy: 0.6750 - v
al_loss: 0.6196 - val_accuracy: 0.6528
Epoch 174/200
44/44 [==============================] - 0s 772us/step - loss: 0.5878 - accuracy: 0.6770 - v
al_loss: 0.6207 - val_accuracy: 0.6571
Epoch 175/200
44/44 [==============================] - 0s 772us/step - loss: 0.5888 - accuracy: 0.6758 - v
al_loss: 0.6250 - val_accuracy: 0.6559
Epoch 176/200
44/44 [==============================] - 0s 795us/step - loss: 0.5885 - accuracy: 0.6759 - v
al_loss: 0.6210 - val_accuracy: 0.6499
Epoch 177/200
44/44 [==============================] - 0s 772us/step - loss: 0.5889 - accuracy: 0.6782 - v
al_loss: 0.6202 - val_accuracy: 0.6534
Epoch 178/200
44/44 [==============================] - 0s 772us/step - loss: 0.5877 - accuracy: 0.6784 - v
al_loss: 0.6196 - val_accuracy: 0.6547
Epoch 179/200
44/44 [==============================] - 0s 795us/step - loss: 0.5883 - accuracy: 0.6795 - v
al_loss: 0.6204 - val_accuracy: 0.6559
Epoch 180/200
44/44 [==============================] - 0s 795us/step - loss: 0.5878 - accuracy: 0.6812 - v
al_loss: 0.6243 - val_accuracy: 0.6470
Epoch 181/200
44/44 [==============================] - 0s 772us/step - loss: 0.5882 - accuracy: 0.6777 - v
al_loss: 0.6209 - val_accuracy: 0.6474
Epoch 182/200
44/44 [==============================] - 0s 772us/step - loss: 0.5881 - accuracy: 0.6792 - v
al_loss: 0.6193 - val_accuracy: 0.6582
Epoch 183/200
44/44 [==============================] - 0s 817us/step - loss: 0.5878 - accuracy: 0.6790 - v
al_loss: 0.6205 - val_accuracy: 0.6484
Epoch 184/200
44/44 [==============================] - 0s 772us/step - loss: 0.5864 - accuracy: 0.6805 - v
al_loss: 0.6201 - val_accuracy: 0.6547
Epoch 185/200
44/44 [==============================] - 0s 795us/step - loss: 0.5861 - accuracy: 0.6799 - v
al_loss: 0.6218 - val_accuracy: 0.6474
Epoch 186/200
44/44 [==============================] - 0s 772us/step - loss: 0.5869 - accuracy: 0.6757 - v
al_loss: 0.6213 - val_accuracy: 0.6507
Epoch 187/200
44/44 [==============================] - 0s 772us/step - loss: 0.5858 - accuracy: 0.6812 - v
al_loss: 0.6218 - val_accuracy: 0.6497
Epoch 188/200
44/44 [==============================] - 0s 772us/step - loss: 0.5861 - accuracy: 0.6801 - v
al_loss: 0.6195 - val_accuracy: 0.6522
Epoch 189/200
44/44 [==============================] - 0s 772us/step - loss: 0.5856 - accuracy: 0.6788 - v
al_loss: 0.6198 - val_accuracy: 0.6536
Epoch 190/200
44/44 [==============================] - 0s 772us/step - loss: 0.5861 - accuracy: 0.6805 - v
al_loss: 0.6189 - val_accuracy: 0.6559
Epoch 191/200
44/44 [==============================] - 0s 772us/step - loss: 0.5853 - accuracy: 0.6797 - v
al_loss: 0.6195 - val_accuracy: 0.6491
```

```
Epoch 192/200
44/44 [==============================] - 0s 772us/step - loss: 0.5846 - accuracy: 0.6818 - v
al_loss: 0.6203 - val_accuracy: 0.6495
Epoch 193/200
44/44 [==============================] - 0s 795us/step - loss: 0.5854 - accuracy: 0.6797 - v
al_loss: 0.6202 - val_accuracy: 0.6453
Epoch 194/200
44/44 [==============================] - 0s 818us/step - loss: 0.5851 - accuracy: 0.6805 - v
al_loss: 0.6186 - val_accuracy: 0.6561
Epoch 195/200
44/44 [==============================] - 0s 772us/step - loss: 0.5850 - accuracy: 0.6792 - v
al_loss: 0.6184 - val_accuracy: 0.6584
Epoch 196/200
44/44 [==============================] - 0s 818us/step - loss: 0.5840 - accuracy: 0.6805 - v
al_loss: 0.6187 - val_accuracy: 0.6524
Epoch 197/200
44/44 [==============================] - 0s 772us/step - loss: 0.5850 - accuracy: 0.6788 - v
al_loss: 0.6196 - val_accuracy: 0.6567
Epoch 198/200
44/44 [==============================] - 0s 795us/step - loss: 0.5845 - accuracy: 0.6786 - v
al_loss: 0.6173 - val_accuracy: 0.6578
Epoch 199/200
44/44 [==============================] - 0s 795us/step - loss: 0.5838 - accuracy: 0.6814 - v
al_loss: 0.6179 - val_accuracy: 0.6565
Epoch 200/200
44/44 [==============================] - 0s 795us/step - loss: 0.5838 - accuracy: 0.6789 - v
al_loss: 0.6186 - val_accuracy: 0.6489
```

Out[67]:

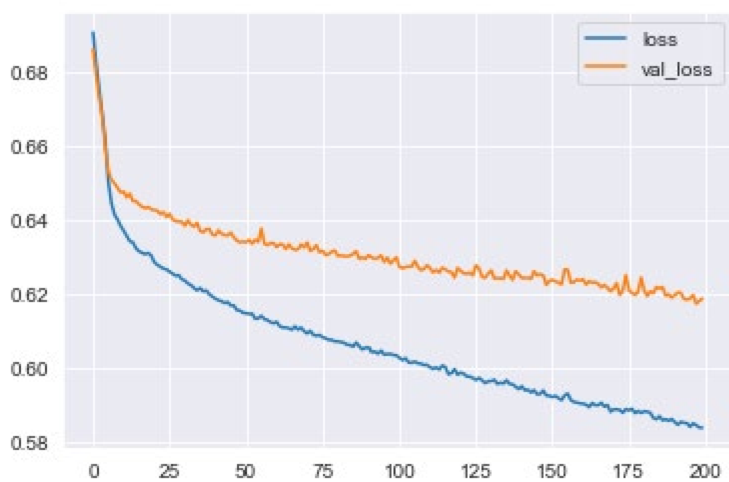
```
<tensorflow.python.keras.callbacks.History at 0x207bb50c850>
```

In [68]:

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

Out[68]:

```
<AxesSubplot:>
```



In [69]:

```
predictions = model.predict_classes(X_test)

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

```
[[1437 1000]
 [ 695 1695]]
              precision    recall  f1-score   support

           0       0.67      0.59      0.63      2437
```

```
           1       0.63      0.71      0.67      2390

    accuracy                           0.65      4827
   macro avg       0.65      0.65      0.65      4827
weighted avg       0.65      0.65      0.65      4827
```

```
C:\Users\User\anaconda3\lib\site-packages\tensorflow\python\keras\engine\sequential.py:455:
UserWarning: `model.predict_classes()` is deprecated and will be removed after 2021-01-01. P
lease use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class
 classification   (e.g. if it uses a `softmax` last-layer activation).* `(model.predict(x) >
 0.5).astype("int32")`,   if your model does binary classification   (e.g. if it uses a `sig
moid` last-layer activation).
  warnings.warn('`model.predict classes()` is deprecated and '
```

In [70]:

```
model_new = Sequential()

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

model_new.add(Dropout(0.2))

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

model_new.add(Dropout(0.2))

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

model_new.add(Dropout(0.2))

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

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


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 [==============================] - 0s 3ms/step - loss: 0.6928 - binary_accuracy: 0.518
6 - val_loss: 0.6886 - val_binary_accuracy: 0.5859
Epoch 2/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6860 - binary_accuracy: 0.550
0 - val_loss: 0.6803 - val_binary_accuracy: 0.5975
```

```
Epoch 3/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6822 - binary_accuracy: 0.565
7 - val_loss: 0.6767 - val_binary_accuracy: 0.6002
Epoch 4/200
44/44 [==============================] - 0s 977us/step - loss: 0.6807 - binary_accuracy: 0.5
700 - val_loss: 0.6741 - val_binary_accuracy: 0.6020
Epoch 5/200
44/44 [==============================] - 0s 977us/step - loss: 0.6770 - binary_accuracy: 0.5
838 - val_loss: 0.6699 - val_binary_accuracy: 0.6093
Epoch 6/200
44/44 [==============================] - 0s 954us/step - loss: 0.6747 - binary_accuracy: 0.5
896 - val_loss: 0.6676 - val_binary_accuracy: 0.6078
Epoch 7/200
44/44 [==============================] - 0s 954us/step - loss: 0.6733 - binary_accuracy: 0.5
899 - val_loss: 0.6650 - val_binary_accuracy: 0.6101
Epoch 8/200
44/44 [==============================] - 0s 977us/step - loss: 0.6697 - binary_accuracy: 0.5
978 - val_loss: 0.6632 - val_binary_accuracy: 0.6111
Epoch 9/200
44/44 [==============================] - 0s 954us/step - loss: 0.6669 - binary_accuracy: 0.5
977 - val_loss: 0.6604 - val_binary_accuracy: 0.6194
Epoch 10/200
44/44 [==============================] - 0s 977us/step - loss: 0.6659 - binary_accuracy: 0.5
991 - val_loss: 0.6597 - val_binary_accuracy: 0.6130
Epoch 11/200
44/44 [==============================] - 0s 954us/step - loss: 0.6629 - binary_accuracy: 0.6
036 - val_loss: 0.6581 - val_binary_accuracy: 0.6149
Epoch 12/200
44/44 [==============================] - 0s 977us/step - loss: 0.6627 - binary_accuracy: 0.6
012 - val_loss: 0.6570 - val_binary_accuracy: 0.6147
Epoch 13/200
44/44 [==============================] - 0s 977us/step - loss: 0.6617 - binary_accuracy: 0.6
035 - val_loss: 0.6569 - val_binary_accuracy: 0.6120
Epoch 14/200
44/44 [==============================] - 0s 977us/step - loss: 0.6607 - binary_accuracy: 0.6
034 - val_loss: 0.6552 - val_binary_accuracy: 0.6130
Epoch 15/200
44/44 [==============================] - 0s 977us/step - loss: 0.6575 - binary_accuracy: 0.6
098 - val_loss: 0.6539 - val_binary_accuracy: 0.6126
Epoch 16/200
44/44 [==============================] - 0s 954us/step - loss: 0.6555 - binary_accuracy: 0.6
108 - val_loss: 0.6530 - val_binary_accuracy: 0.6097
Epoch 17/200
44/44 [==============================] - 0s 977us/step - loss: 0.6586 - binary_accuracy: 0.6
015 - val_loss: 0.6531 - val_binary_accuracy: 0.6099
Epoch 18/200
44/44 [==============================] - 0s 977us/step - loss: 0.6544 - binary_accuracy: 0.6
083 - val_loss: 0.6514 - val_binary_accuracy: 0.6124
Epoch 19/200
44/44 [==============================] - 0s 976us/step - loss: 0.6546 - binary_accuracy: 0.6
103 - val_loss: 0.6517 - val_binary_accuracy: 0.6111
Epoch 20/200
44/44 [==============================] - 0s 977us/step - loss: 0.6521 - binary_accuracy: 0.6
108 - val_loss: 0.6502 - val_binary_accuracy: 0.6190
Epoch 21/200
44/44 [==============================] - 0s 977us/step - loss: 0.6527 - binary_accuracy: 0.6
132 - val_loss: 0.6504 - val_binary_accuracy: 0.6196
Epoch 22/200
44/44 [==============================] - 0s 977us/step - loss: 0.6493 - binary_accuracy: 0.6
130 - val_loss: 0.6489 - val_binary_accuracy: 0.6174
Epoch 23/200
44/44 [==============================] - 0s 954us/step - loss: 0.6518 - binary_accuracy: 0.6
148 - val_loss: 0.6485 - val_binary_accuracy: 0.6172
Epoch 24/200
44/44 [==============================] - 0s 954us/step - loss: 0.6490 - binary_accuracy: 0.6
147 - val_loss: 0.6483 - val_binary_accuracy: 0.6203
Epoch 25/200
44/44 [==============================] - 0s 954us/step - loss: 0.6534 - binary_accuracy: 0.6
128 - val_loss: 0.6477 - val_binary_accuracy: 0.6155
```

```
Epoch 26/200
44/44 [==============================] - 0s 977us/step - loss: 0.6487 - binary_accuracy: 0.6
163 - val_loss: 0.6471 - val_binary_accuracy: 0.6196
Epoch 27/200
44/44 [==============================] - 0s 954us/step - loss: 0.6512 - binary_accuracy: 0.6
164 - val_loss: 0.6472 - val_binary_accuracy: 0.6151
Epoch 28/200
44/44 [==============================] - 0s 954us/step - loss: 0.6484 - binary_accuracy: 0.6
146 - val_loss: 0.6465 - val_binary_accuracy: 0.6147
Epoch 29/200
44/44 [==============================] - 0s 976us/step - loss: 0.6522 - binary_accuracy: 0.6
138 - val_loss: 0.6461 - val_binary_accuracy: 0.6192
Epoch 30/200
44/44 [==============================] - 0s 977us/step - loss: 0.6459 - binary_accuracy: 0.6
222 - val_loss: 0.6453 - val_binary_accuracy: 0.6165
Epoch 31/200
44/44 [==============================] - 0s 976us/step - loss: 0.6472 - binary_accuracy: 0.6
160 - val_loss: 0.6445 - val_binary_accuracy: 0.6205
Epoch 32/200
44/44 [==============================] - 0s 999us/step - loss: 0.6472 - binary_accuracy: 0.6
197 - val_loss: 0.6440 - val_binary_accuracy: 0.6259
Epoch 33/200
44/44 [==============================] - 0s 954us/step - loss: 0.6461 - binary_accuracy: 0.6
205 - val_loss: 0.6438 - val_binary_accuracy: 0.6256
Epoch 34/200
44/44 [==============================] - 0s 931us/step - loss: 0.6472 - binary_accuracy: 0.6
179 - val_loss: 0.6429 - val_binary_accuracy: 0.6275
Epoch 35/200
44/44 [==============================] - 0s 954us/step - loss: 0.6456 - binary_accuracy: 0.6
216 - val_loss: 0.6436 - val_binary_accuracy: 0.6201
Epoch 36/200
44/44 [==============================] - 0s 931us/step - loss: 0.6460 - binary_accuracy: 0.6
250 - val_loss: 0.6430 - val_binary_accuracy: 0.6242
Epoch 37/200
44/44 [==============================] - 0s 954us/step - loss: 0.6454 - binary_accuracy: 0.6
218 - val_loss: 0.6434 - val_binary_accuracy: 0.6256
Epoch 38/200
44/44 [==============================] - 0s 954us/step - loss: 0.6471 - binary_accuracy: 0.6
192 - val_loss: 0.6426 - val_binary_accuracy: 0.6186
Epoch 39/200
44/44 [==============================] - 0s 954us/step - loss: 0.6451 - binary_accuracy: 0.6
209 - val_loss: 0.6420 - val_binary_accuracy: 0.6277
Epoch 40/200
44/44 [==============================] - 0s 954us/step - loss: 0.6464 - binary_accuracy: 0.6
245 - val_loss: 0.6420 - val_binary_accuracy: 0.6246
Epoch 41/200
44/44 [==============================] - 0s 999us/step - loss: 0.6424 - binary_accuracy: 0.6
292 - val_loss: 0.6411 - val_binary_accuracy: 0.6219
Epoch 42/200
44/44 [==============================] - 0s 954us/step - loss: 0.6436 - binary_accuracy: 0.6
219 - val_loss: 0.6414 - val_binary_accuracy: 0.6279
Epoch 43/200
44/44 [==============================] - 0s 977us/step - loss: 0.6430 - binary_accuracy: 0.6
256 - val_loss: 0.6407 - val_binary_accuracy: 0.6273
Epoch 44/200
44/44 [==============================] - 0s 954us/step - loss: 0.6425 - binary_accuracy: 0.6
275 - val_loss: 0.6403 - val_binary_accuracy: 0.6242
Epoch 45/200
44/44 [==============================] - 0s 977us/step - loss: 0.6443 - binary_accuracy: 0.6
235 - val_loss: 0.6407 - val_binary_accuracy: 0.6296
Epoch 46/200
44/44 [==============================] - 0s 977us/step - loss: 0.6410 - binary_accuracy: 0.6
280 - val_loss: 0.6401 - val_binary_accuracy: 0.6285
Epoch 47/200
44/44 [==============================] - 0s 976us/step - loss: 0.6456 - binary_accuracy: 0.6
256 - val_loss: 0.6401 - val_binary_accuracy: 0.6294
Epoch 48/200
44/44 [==============================] - 0s 954us/step - loss: 0.6422 - binary_accuracy: 0.6
283 - val_loss: 0.6394 - val_binary_accuracy: 0.6302
```

```
Epoch 49/200
44/44 [==============================] - 0s 954us/step - loss: 0.6408 - binary_accuracy: 0.6
235 - val_loss: 0.6394 - val_binary_accuracy: 0.6273
Epoch 50/200
44/44 [==============================] - 0s 977us/step - loss: 0.6375 - binary_accuracy: 0.6
323 - val_loss: 0.6381 - val_binary_accuracy: 0.6304
Epoch 51/200
44/44 [==============================] - 0s 976us/step - loss: 0.6429 - binary_accuracy: 0.6
296 - val_loss: 0.6385 - val_binary_accuracy: 0.6244
Epoch 52/200
44/44 [==============================] - 0s 977us/step - loss: 0.6409 - binary_accuracy: 0.6
278 - val_loss: 0.6380 - val_binary_accuracy: 0.6230
Epoch 53/200
44/44 [==============================] - 0s 977us/step - loss: 0.6377 - binary_accuracy: 0.6
313 - val_loss: 0.6388 - val_binary_accuracy: 0.6302
Epoch 54/200
44/44 [==============================] - 0s 977us/step - loss: 0.6426 - binary_accuracy: 0.6
229 - val_loss: 0.6371 - val_binary_accuracy: 0.6298
Epoch 55/200
44/44 [==============================] - 0s 976us/step - loss: 0.6383 - binary_accuracy: 0.6
284 - val_loss: 0.6373 - val_binary_accuracy: 0.6302
Epoch 56/200
44/44 [==============================] - 0s 976us/step - loss: 0.6375 - binary_accuracy: 0.6
339 - val_loss: 0.6364 - val_binary_accuracy: 0.6250
Epoch 57/200
44/44 [==============================] - 0s 977us/step - loss: 0.6389 - binary_accuracy: 0.6
366 - val_loss: 0.6372 - val_binary_accuracy: 0.6252
Epoch 58/200
44/44 [==============================] - 0s 977us/step - loss: 0.6386 - binary_accuracy: 0.6
321 - val_loss: 0.6364 - val_binary_accuracy: 0.6259
Epoch 59/200
44/44 [==============================] - 0s 954us/step - loss: 0.6380 - binary_accuracy: 0.6
334 - val_loss: 0.6359 - val_binary_accuracy: 0.6292
Epoch 60/200
44/44 [==============================] - 0s 977us/step - loss: 0.6362 - binary_accuracy: 0.6
330 - val_loss: 0.6359 - val_binary_accuracy: 0.6296
Epoch 61/200
44/44 [==============================] - 0s 977us/step - loss: 0.6347 - binary_accuracy: 0.6
328 - val_loss: 0.6356 - val_binary_accuracy: 0.6308
Epoch 62/200
44/44 [==============================] - 0s 954us/step - loss: 0.6376 - binary_accuracy: 0.6
317 - val_loss: 0.6358 - val_binary_accuracy: 0.6339
Epoch 63/200
44/44 [==============================] - 0s 954us/step - loss: 0.6359 - binary_accuracy: 0.6
313 - val_loss: 0.6351 - val_binary_accuracy: 0.6323
Epoch 64/200
44/44 [==============================] - 0s 954us/step - loss: 0.6376 - binary_accuracy: 0.6
322 - val_loss: 0.6355 - val_binary_accuracy: 0.6294
Epoch 65/200
44/44 [==============================] - 0s 977us/step - loss: 0.6353 - binary_accuracy: 0.6
300 - val_loss: 0.6357 - val_binary_accuracy: 0.6364
Epoch 66/200
44/44 [==============================] - 0s 977us/step - loss: 0.6365 - binary_accuracy: 0.6
324 - val_loss: 0.6341 - val_binary_accuracy: 0.6358
Epoch 67/200
44/44 [==============================] - 0s 976us/step - loss: 0.6348 - binary_accuracy: 0.6
339 - val_loss: 0.6359 - val_binary_accuracy: 0.6198
Epoch 68/200
44/44 [==============================] - 0s 954us/step - loss: 0.6339 - binary_accuracy: 0.6
391 - val_loss: 0.6338 - val_binary_accuracy: 0.6375
Epoch 69/200
44/44 [==============================] - 0s 977us/step - loss: 0.6387 - binary_accuracy: 0.6
315 - val_loss: 0.6338 - val_binary_accuracy: 0.6360
Epoch 70/200
44/44 [==============================] - 0s 977us/step - loss: 0.6340 - binary_accuracy: 0.6
330 - val_loss: 0.6328 - val_binary_accuracy: 0.6350
Epoch 71/200
44/44 [==============================] - 0s 954us/step - loss: 0.6331 - binary_accuracy: 0.6
371 - val_loss: 0.6337 - val_binary_accuracy: 0.6354
```

```
Epoch 72/200
44/44 [==============================] - 0s 954us/step - loss: 0.6342 - binary_accuracy: 0.6
322 - val_loss: 0.6325 - val_binary_accuracy: 0.6385
Epoch 73/200
44/44 [==============================] - 0s 977us/step - loss: 0.6334 - binary_accuracy: 0.6
345 - val_loss: 0.6327 - val_binary_accuracy: 0.6366
Epoch 74/200
44/44 [==============================] - 0s 977us/step - loss: 0.6329 - binary_accuracy: 0.6
393 - val_loss: 0.6329 - val_binary_accuracy: 0.6379
Epoch 75/200
44/44 [==============================] - 0s 977us/step - loss: 0.6348 - binary_accuracy: 0.6
361 - val_loss: 0.6322 - val_binary_accuracy: 0.6408
Epoch 76/200
44/44 [==============================] - 0s 954us/step - loss: 0.6316 - binary_accuracy: 0.6
402 - val_loss: 0.6321 - val_binary_accuracy: 0.6352
Epoch 77/200
44/44 [==============================] - 0s 977us/step - loss: 0.6327 - binary_accuracy: 0.6
367 - val_loss: 0.6332 - val_binary_accuracy: 0.6410
Epoch 78/200
44/44 [==============================] - 0s 999us/step - loss: 0.6346 - binary_accuracy: 0.6
397 - val_loss: 0.6323 - val_binary_accuracy: 0.6395
Epoch 79/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6351 - binary_accuracy: 0.633
0 - val_loss: 0.6321 - val_binary_accuracy: 0.6370
Epoch 80/200
44/44 [==============================] - 0s 999us/step - loss: 0.6324 - binary_accuracy: 0.6
387 - val_loss: 0.6322 - val_binary_accuracy: 0.6339
Epoch 81/200
44/44 [==============================] - 0s 999us/step - loss: 0.6314 - binary_accuracy: 0.6
392 - val_loss: 0.6313 - val_binary_accuracy: 0.6420
Epoch 82/200
44/44 [==============================] - 0s 977us/step - loss: 0.6312 - binary_accuracy: 0.6
412 - val_loss: 0.6312 - val_binary_accuracy: 0.6406
Epoch 83/200
44/44 [==============================] - 0s 953us/step - loss: 0.6292 - binary_accuracy: 0.6
442 - val_loss: 0.6305 - val_binary_accuracy: 0.6397
Epoch 84/200
44/44 [==============================] - 0s 954us/step - loss: 0.6279 - binary_accuracy: 0.6
417 - val_loss: 0.6306 - val_binary_accuracy: 0.6439
Epoch 85/200
44/44 [==============================] - 0s 977us/step - loss: 0.6329 - binary_accuracy: 0.6
358 - val_loss: 0.6306 - val_binary_accuracy: 0.6443
Epoch 86/200
44/44 [==============================] - 0s 999us/step - loss: 0.6309 - binary_accuracy: 0.6
420 - val_loss: 0.6308 - val_binary_accuracy: 0.6430
Epoch 87/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6314 - binary_accuracy: 0.640
1 - val_loss: 0.6304 - val_binary_accuracy: 0.6445
Epoch 88/200
44/44 [==============================] - 0s 954us/step - loss: 0.6288 - binary_accuracy: 0.6
441 - val_loss: 0.6303 - val_binary_accuracy: 0.6437
Epoch 89/200
44/44 [==============================] - 0s 954us/step - loss: 0.6278 - binary_accuracy: 0.6
443 - val_loss: 0.6305 - val_binary_accuracy: 0.6410
Epoch 90/200
44/44 [==============================] - 0s 954us/step - loss: 0.6296 - binary_accuracy: 0.6
417 - val_loss: 0.6294 - val_binary_accuracy: 0.6428
Epoch 91/200
44/44 [==============================] - 0s 954us/step - loss: 0.6278 - binary_accuracy: 0.6
460 - val_loss: 0.6296 - val_binary_accuracy: 0.6433
Epoch 92/200
44/44 [==============================] - 0s 954us/step - loss: 0.6276 - binary_accuracy: 0.6
452 - val_loss: 0.6299 - val_binary_accuracy: 0.6472
Epoch 93/200
44/44 [==============================] - 0s 954us/step - loss: 0.6299 - binary_accuracy: 0.6
432 - val_loss: 0.6293 - val_binary_accuracy: 0.6468
Epoch 94/200
44/44 [==============================] - 0s 954us/step - loss: 0.6273 - binary_accuracy: 0.6
417 - val_loss: 0.6301 - val_binary_accuracy: 0.6399
```

```
Epoch 95/200
44/44 [==============================] - 0s 954us/step - loss: 0.6253 - binary_accuracy: 0.6
488 - val_loss: 0.6288 - val_binary_accuracy: 0.6455
Epoch 96/200
44/44 [==============================] - 0s 954us/step - loss: 0.6293 - binary_accuracy: 0.6
428 - val_loss: 0.6288 - val_binary_accuracy: 0.6420
Epoch 97/200
44/44 [==============================] - 0s 954us/step - loss: 0.6289 - binary_accuracy: 0.6
445 - val_loss: 0.6278 - val_binary_accuracy: 0.6470
Epoch 98/200
44/44 [==============================] - 0s 977us/step - loss: 0.6243 - binary_accuracy: 0.6
498 - val_loss: 0.6280 - val_binary_accuracy: 0.6468
Epoch 99/200
44/44 [==============================] - 0s 976us/step - loss: 0.6287 - binary_accuracy: 0.6
474 - val_loss: 0.6290 - val_binary_accuracy: 0.6404
Epoch 100/200
44/44 [==============================] - 0s 976us/step - loss: 0.6240 - binary_accuracy: 0.6
459 - val_loss: 0.6273 - val_binary_accuracy: 0.6497
Epoch 101/200
44/44 [==============================] - 0s 976us/step - loss: 0.6297 - binary_accuracy: 0.6
438 - val_loss: 0.6276 - val_binary_accuracy: 0.6515
Epoch 102/200
44/44 [==============================] - 0s 954us/step - loss: 0.6289 - binary_accuracy: 0.6
452 - val_loss: 0.6269 - val_binary_accuracy: 0.6513
Epoch 103/200
44/44 [==============================] - 0s 977us/step - loss: 0.6253 - binary_accuracy: 0.6
441 - val_loss: 0.6270 - val_binary_accuracy: 0.6437
Epoch 104/200
44/44 [==============================] - 0s 954us/step - loss: 0.6237 - binary_accuracy: 0.6
455 - val_loss: 0.6266 - val_binary_accuracy: 0.6505
Epoch 105/200
44/44 [==============================] - 0s 954us/step - loss: 0.6240 - binary_accuracy: 0.6
493 - val_loss: 0.6264 - val_binary_accuracy: 0.6489
Epoch 106/200
44/44 [==============================] - 0s 953us/step - loss: 0.6252 - binary_accuracy: 0.6
458 - val_loss: 0.6263 - val_binary_accuracy: 0.6522
Epoch 107/200
44/44 [==============================] - 0s 954us/step - loss: 0.6284 - binary_accuracy: 0.6
463 - val_loss: 0.6269 - val_binary_accuracy: 0.6466
Epoch 108/200
44/44 [==============================] - 0s 977us/step - loss: 0.6227 - binary_accuracy: 0.6
541 - val_loss: 0.6255 - val_binary_accuracy: 0.6474
Epoch 109/200
44/44 [==============================] - 0s 954us/step - loss: 0.6266 - binary_accuracy: 0.6
469 - val_loss: 0.6266 - val_binary_accuracy: 0.6449
Epoch 110/200
44/44 [==============================] - 0s 976us/step - loss: 0.6248 - binary_accuracy: 0.6
477 - val_loss: 0.6258 - val_binary_accuracy: 0.6513
Epoch 111/200
44/44 [==============================] - 0s 999us/step - loss: 0.6267 - binary_accuracy: 0.6
435 - val_loss: 0.6257 - val_binary_accuracy: 0.6495
Epoch 112/200
44/44 [==============================] - 0s 931us/step - loss: 0.6248 - binary_accuracy: 0.6
493 - val_loss: 0.6258 - val_binary_accuracy: 0.6476
Epoch 113/200
44/44 [==============================] - 0s 977us/step - loss: 0.6220 - binary_accuracy: 0.6
551 - val_loss: 0.6252 - val_binary_accuracy: 0.6520
Epoch 114/200
44/44 [==============================] - 0s 976us/step - loss: 0.6221 - binary_accuracy: 0.6
481 - val_loss: 0.6253 - val_binary_accuracy: 0.6499
Epoch 115/200
44/44 [==============================] - 0s 976us/step - loss: 0.6245 - binary_accuracy: 0.6
453 - val_loss: 0.6255 - val_binary_accuracy: 0.6509
Epoch 116/200
44/44 [==============================] - 0s 999us/step - loss: 0.6230 - binary_accuracy: 0.6
489 - val_loss: 0.6245 - val_binary_accuracy: 0.6547
Epoch 117/200
44/44 [==============================] - 0s 954us/step - loss: 0.6242 - binary_accuracy: 0.6
496 - val_loss: 0.6244 - val_binary_accuracy: 0.6507
```

```
Epoch 118/200
44/44 [==============================] - 0s 977us/step - loss: 0.6226 - binary_accuracy: 0.6
474 - val_loss: 0.6258 - val_binary_accuracy: 0.6509
Epoch 119/200
44/44 [==============================] - 0s 954us/step - loss: 0.6256 - binary_accuracy: 0.6
445 - val_loss: 0.6247 - val_binary_accuracy: 0.6526
Epoch 120/200
44/44 [==============================] - 0s 954us/step - loss: 0.6245 - binary_accuracy: 0.6
461 - val_loss: 0.6242 - val_binary_accuracy: 0.6524
Epoch 121/200
44/44 [==============================] - 0s 954us/step - loss: 0.6244 - binary_accuracy: 0.6
502 - val_loss: 0.6248 - val_binary_accuracy: 0.6453
Epoch 122/200
44/44 [==============================] - 0s 999us/step - loss: 0.6243 - binary_accuracy: 0.6
465 - val_loss: 0.6246 - val_binary_accuracy: 0.6486
Epoch 123/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6243 - binary_accuracy: 0.652
2 - val_loss: 0.6242 - val_binary_accuracy: 0.6542
Epoch 124/200
44/44 [==============================] - 0s 999us/step - loss: 0.6275 - binary_accuracy: 0.6
433 - val_loss: 0.6246 - val_binary_accuracy: 0.6509
Epoch 125/200
44/44 [==============================] - 0s 954us/step - loss: 0.6260 - binary_accuracy: 0.6
488 - val_loss: 0.6247 - val_binary_accuracy: 0.6563
Epoch 126/200
44/44 [==============================] - 0s 954us/step - loss: 0.6223 - binary_accuracy: 0.6
513 - val_loss: 0.6244 - val_binary_accuracy: 0.6486
Epoch 127/200
44/44 [==============================] - 0s 999us/step - loss: 0.6253 - binary_accuracy: 0.6
474 - val_loss: 0.6260 - val_binary_accuracy: 0.6443
Epoch 128/200
44/44 [==============================] - 0s 1ms/step - loss: 0.6213 - binary_accuracy: 0.647
3 - val_loss: 0.6235 - val_binary_accuracy: 0.6561
Epoch 129/200
44/44 [==============================] - 0s 954us/step - loss: 0.6238 - binary_accuracy: 0.6
494 - val_loss: 0.6239 - val_binary_accuracy: 0.6497
Epoch 130/200
44/44 [==============================] - 0s 954us/step - loss: 0.6205 - binary_accuracy: 0.6
527 - val_loss: 0.6238 - val_binary_accuracy: 0.6499
Epoch 131/200
44/44 [==============================] - 0s 954us/step - loss: 0.6213 - binary_accuracy: 0.6
512 - val_loss: 0.6244 - val_binary_accuracy: 0.6518
Epoch 132/200
44/44 [==============================] - 0s 977us/step - loss: 0.6260 - binary_accuracy: 0.6
488 - val_loss: 0.6242 - val_binary_accuracy: 0.6505
Epoch 133/200
44/44 [==============================] - 0s 954us/step - loss: 0.6210 - binary_accuracy: 0.6
501 - val_loss: 0.6230 - val_binary_accuracy: 0.6555
Epoch 134/200
44/44 [==============================] - 0s 977us/step - loss: 0.6206 - binary_accuracy: 0.6
473 - val_loss: 0.6243 - val_binary_accuracy: 0.6520
Epoch 135/200
44/44 [==============================] - 0s 954us/step - loss: 0.6218 - binary_accuracy: 0.6
472 - val_loss: 0.6240 - val_binary_accuracy: 0.6466
Epoch 136/200
44/44 [==============================] - 0s 977us/step - loss: 0.6221 - binary_accuracy: 0.6
521 - val_loss: 0.6228 - val_binary_accuracy: 0.6515
Epoch 137/200
44/44 [==============================] - 0s 954us/step - loss: 0.6236 - binary_accuracy: 0.6
530 - val_loss: 0.6239 - val_binary_accuracy: 0.6474
Epoch 138/200
44/44 [==============================] - 0s 977us/step - loss: 0.6218 - binary_accuracy: 0.6
518 - val_loss: 0.6228 - val_binary_accuracy: 0.6499
Epoch 139/200
44/44 [==============================] - 0s 954us/step - loss: 0.6192 - binary_accuracy: 0.6
552 - val_loss: 0.6232 - val_binary_accuracy: 0.6491
Epoch 140/200
44/44 [==============================] - 0s 954us/step - loss: 0.6196 - binary_accuracy: 0.6
504 - val_loss: 0.6227 - val_binary_accuracy: 0.6497
```

```
Epoch 141/200
44/44 [==============================] - 0s 977us/step - loss: 0.6213 - binary_accuracy: 0.6
504 - val_loss: 0.6226 - val_binary_accuracy: 0.6532
Epoch 142/200
44/44 [==============================] - 0s 977us/step - loss: 0.6218 - binary_accuracy: 0.6
506 - val_loss: 0.6229 - val_binary_accuracy: 0.6497
Epoch 143/200
44/44 [==============================] - 0s 954us/step - loss: 0.6186 - binary_accuracy: 0.6
519 - val_loss: 0.6232 - val_binary_accuracy: 0.6468
Epoch 144/200
44/44 [==============================] - 0s 977us/step - loss: 0.6190 - binary_accuracy: 0.6
547 - val_loss: 0.6221 - val_binary_accuracy: 0.6505
Epoch 145/200
44/44 [==============================] - 0s 954us/step - loss: 0.6205 - binary_accuracy: 0.6
562 - val_loss: 0.6225 - val_binary_accuracy: 0.6551
Epoch 146/200
44/44 [==============================] - 0s 954us/step - loss: 0.6198 - binary_accuracy: 0.6
573 - val_loss: 0.6223 - val_binary_accuracy: 0.6559
Epoch 147/200
44/44 [==============================] - 0s 954us/step - loss: 0.6219 - binary_accuracy: 0.6
492 - val_loss: 0.6220 - val_binary_accuracy: 0.6542
Epoch 148/200
44/44 [==============================] - 0s 954us/step - loss: 0.6188 - binary_accuracy: 0.6
544 - val_loss: 0.6222 - val_binary_accuracy: 0.6578
Epoch 149/200
44/44 [==============================] - 0s 954us/step - loss: 0.6235 - binary_accuracy: 0.6
520 - val_loss: 0.6218 - val_binary_accuracy: 0.6567
Epoch 150/200
44/44 [==============================] - 0s 931us/step - loss: 0.6216 - binary_accuracy: 0.6
497 - val_loss: 0.6214 - val_binary_accuracy: 0.6576
Epoch 151/200
44/44 [==============================] - 0s 954us/step - loss: 0.6211 - binary_accuracy: 0.6
489 - val_loss: 0.6214 - val_binary_accuracy: 0.6598
Epoch 152/200
44/44 [==============================] - 0s 954us/step - loss: 0.6217 - binary_accuracy: 0.6
503 - val_loss: 0.6211 - val_binary_accuracy: 0.6586
Epoch 153/200
44/44 [==============================] - 0s 954us/step - loss: 0.6212 - binary_accuracy: 0.6
488 - val_loss: 0.6217 - val_binary_accuracy: 0.6569
Epoch 154/200
44/44 [==============================] - 0s 977us/step - loss: 0.6183 - binary_accuracy: 0.6
543 - val_loss: 0.6220 - val_binary_accuracy: 0.6486
Epoch 155/200
44/44 [==============================] - 0s 977us/step - loss: 0.6198 - binary_accuracy: 0.6
576 - val_loss: 0.6210 - val_binary_accuracy: 0.6536
Epoch 156/200
44/44 [==============================] - 0s 977us/step - loss: 0.6163 - binary_accuracy: 0.6
578 - val_loss: 0.6211 - val_binary_accuracy: 0.6518
Epoch 157/200
44/44 [==============================] - 0s 977us/step - loss: 0.6193 - binary_accuracy: 0.6
528 - val_loss: 0.6213 - val_binary_accuracy: 0.6551
Epoch 158/200
44/44 [==============================] - 0s 954us/step - loss: 0.6195 - binary_accuracy: 0.6
528 - val_loss: 0.6206 - val_binary_accuracy: 0.6557
Epoch 159/200
44/44 [==============================] - 0s 954us/step - loss: 0.6202 - binary_accuracy: 0.6
534 - val_loss: 0.6206 - val_binary_accuracy: 0.6532
Epoch 160/200
44/44 [==============================] - 0s 954us/step - loss: 0.6188 - binary_accuracy: 0.6
525 - val_loss: 0.6203 - val_binary_accuracy: 0.6563
Epoch 161/200
44/44 [==============================] - 0s 977us/step - loss: 0.6188 - binary_accuracy: 0.6
497 - val_loss: 0.6208 - val_binary_accuracy: 0.6538
Epoch 162/200
44/44 [==============================] - 0s 954us/step - loss: 0.6190 - binary_accuracy: 0.6
520 - val_loss: 0.6213 - val_binary_accuracy: 0.6578
Epoch 163/200
44/44 [==============================] - 0s 954us/step - loss: 0.6185 - binary_accuracy: 0.6
547 - val_loss: 0.6205 - val_binary_accuracy: 0.6549
```

```
Epoch 164/200
44/44 [==============================] - 0s 954us/step - loss: 0.6182 - binary_accuracy: 0.6
522 - val_loss: 0.6203 - val_binary_accuracy: 0.6578
Epoch 165/200
44/44 [==============================] - 0s 954us/step - loss: 0.6201 - binary_accuracy: 0.6
536 - val_loss: 0.6203 - val_binary_accuracy: 0.6580
Epoch 166/200
44/44 [==============================] - 0s 954us/step - loss: 0.6196 - binary_accuracy: 0.6
505 - val_loss: 0.6199 - val_binary_accuracy: 0.6536
Epoch 167/200
44/44 [==============================] - 0s 954us/step - loss: 0.6200 - binary_accuracy: 0.6
533 - val_loss: 0.6212 - val_binary_accuracy: 0.6536
Epoch 168/200
44/44 [==============================] - 0s 954us/step - loss: 0.6161 - binary_accuracy: 0.6
565 - val_loss: 0.6203 - val_binary_accuracy: 0.6584
Epoch 169/200
44/44 [==============================] - 0s 954us/step - loss: 0.6196 - binary_accuracy: 0.6
506 - val_loss: 0.6213 - val_binary_accuracy: 0.6513
Epoch 170/200
44/44 [==============================] - 0s 954us/step - loss: 0.6186 - binary_accuracy: 0.6
537 - val_loss: 0.6206 - val_binary_accuracy: 0.6540
Epoch 171/200
44/44 [==============================] - 0s 954us/step - loss: 0.6195 - binary_accuracy: 0.6
520 - val_loss: 0.6204 - val_binary_accuracy: 0.6555
Epoch 172/200
44/44 [==============================] - 0s 977us/step - loss: 0.6173 - binary_accuracy: 0.6
547 - val_loss: 0.6206 - val_binary_accuracy: 0.6580
Epoch 173/200
44/44 [==============================] - 0s 954us/step - loss: 0.6171 - binary_accuracy: 0.6
553 - val_loss: 0.6197 - val_binary_accuracy: 0.6551
Epoch 174/200
44/44 [==============================] - 0s 954us/step - loss: 0.6171 - binary_accuracy: 0.6
512 - val_loss: 0.6194 - val_binary_accuracy: 0.6559
Epoch 175/200
44/44 [==============================] - 0s 954us/step - loss: 0.6143 - binary_accuracy: 0.6
572 - val_loss: 0.6195 - val_binary_accuracy: 0.6540
Epoch 176/200
44/44 [==============================] - 0s 954us/step - loss: 0.6207 - binary_accuracy: 0.6
507 - val_loss: 0.6199 - val_binary_accuracy: 0.6505
Epoch 177/200
44/44 [==============================] - 0s 954us/step - loss: 0.6166 - binary_accuracy: 0.6
607 - val_loss: 0.6193 - val_binary_accuracy: 0.6573
Epoch 178/200
44/44 [==============================] - 0s 954us/step - loss: 0.6155 - binary_accuracy: 0.6
564 - val_loss: 0.6201 - val_binary_accuracy: 0.6555
Epoch 179/200
44/44 [==============================] - 0s 954us/step - loss: 0.6175 - binary_accuracy: 0.6
509 - val_loss: 0.6194 - val_binary_accuracy: 0.6565
Epoch 180/200
44/44 [==============================] - 0s 954us/step - loss: 0.6194 - binary_accuracy: 0.6
513 - val_loss: 0.6186 - val_binary_accuracy: 0.6598
Epoch 181/200
44/44 [==============================] - 0s 999us/step - loss: 0.6218 - binary_accuracy: 0.6
530 - val_loss: 0.6194 - val_binary_accuracy: 0.6555
Epoch 182/200
44/44 [==============================] - 0s 954us/step - loss: 0.6201 - binary_accuracy: 0.6
488 - val_loss: 0.6197 - val_binary_accuracy: 0.6542
Epoch 183/200
44/44 [==============================] - 0s 976us/step - loss: 0.6172 - binary_accuracy: 0.6
552 - val_loss: 0.6184 - val_binary_accuracy: 0.6594
Epoch 184/200
44/44 [==============================] - 0s 976us/step - loss: 0.6159 - binary_accuracy: 0.6
590 - val_loss: 0.6205 - val_binary_accuracy: 0.6547
Epoch 185/200
44/44 [==============================] - 0s 954us/step - loss: 0.6184 - binary_accuracy: 0.6
536 - val_loss: 0.6182 - val_binary_accuracy: 0.6613
Epoch 186/200
44/44 [==============================] - 0s 954us/step - loss: 0.6140 - binary_accuracy: 0.6
610 - val_loss: 0.6183 - val_binary_accuracy: 0.6598
```

```
Epoch 187/200
44/44 [==============================] - 0s 977us/step - loss: 0.6147 - binary_accuracy: 0.6
589 - val_loss: 0.6187 - val_binary_accuracy: 0.6549
Epoch 188/200
44/44 [==============================] - 0s 954us/step - loss: 0.6116 - binary_accuracy: 0.6
603 - val_loss: 0.6186 - val_binary_accuracy: 0.6602
Epoch 189/200
44/44 [==============================] - 0s 954us/step - loss: 0.6166 - binary_accuracy: 0.6
576 - val_loss: 0.6186 - val_binary_accuracy: 0.6584
Epoch 190/200
44/44 [==============================] - 0s 977us/step - loss: 0.6172 - binary_accuracy: 0.6
526 - val_loss: 0.6193 - val_binary_accuracy: 0.6547
Epoch 191/200
44/44 [==============================] - 0s 954us/step - loss: 0.6149 - binary_accuracy: 0.6
557 - val_loss: 0.6182 - val_binary_accuracy: 0.6532
Epoch 192/200
44/44 [==============================] - 0s 977us/step - loss: 0.6177 - binary_accuracy: 0.6
521 - val_loss: 0.6176 - val_binary_accuracy: 0.6567
Epoch 193/200
44/44 [==============================] - 0s 954us/step - loss: 0.6189 - binary_accuracy: 0.6
522 - val_loss: 0.6179 - val_binary_accuracy: 0.6584
Epoch 194/200
44/44 [==============================] - 0s 977us/step - loss: 0.6129 - binary_accuracy: 0.6
598 - val_loss: 0.6177 - val_binary_accuracy: 0.6600
Epoch 195/200
44/44 [==============================] - 0s 977us/step - loss: 0.6146 - binary_accuracy: 0.6
572 - val_loss: 0.6176 - val_binary_accuracy: 0.6559
Epoch 196/200
44/44 [==============================] - 0s 954us/step - loss: 0.6164 - binary_accuracy: 0.6
516 - val_loss: 0.6176 - val_binary_accuracy: 0.6619
Epoch 197/200
44/44 [==============================] - 0s 977us/step - loss: 0.6161 - binary_accuracy: 0.6
582 - val_loss: 0.6178 - val_binary_accuracy: 0.6580
Epoch 198/200
44/44 [==============================] - 0s 977us/step - loss: 0.6147 - binary_accuracy: 0.6
573 - val_loss: 0.6187 - val_binary_accuracy: 0.6596
Epoch 199/200
44/44 [==============================] - 0s 976us/step - loss: 0.6174 - binary_accuracy: 0.6
605 - val_loss: 0.6182 - val_binary_accuracy: 0.6540
Epoch 200/200
44/44 [==============================] - 0s 977us/step - loss: 0.6170 - binary_accuracy: 0.6
524 - val_loss: 0.6175 - val_binary_accuracy: 0.6582
```

Out[70]:

```
<tensorflow.python.keras.callbacks.History at 0x207c3902f10>
```

In [71]:

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

Out[71]:

```
<AxesSubplot:>
```

In [72]:

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

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

```
[[ 242 2195]
 [  30 2360]]
              precision    recall  f1-score   support

           0       0.89      0.10      0.18      2437
           1       0.52      0.99      0.68      2390

    accuracy                           0.54      4827
   macro avg       0.70      0.54      0.43      4827
weighted avg       0.71      0.54      0.43      4827

C:\Users\User\anaconda3\lib\site-packages\tensorflow\python\keras\engine\sequential.py:430:
UserWarning: `model.predict_proba()` is deprecated and will be removed after 2021-01-01. Ple
ase use `model.predict()` instead.
  warnings.warn('`model.predict_proba()` is deprecated and '
```

By : Abdullah Alwabel