Sepsis_2

April 8, 2023

1 Early Prediction of Sepsis from Clinical Data:

2 the PhysioNet/Computing in Cardiology Challenge 2019

8 4 2023 K. Kh

[ ]:
```
from keras.layers import Dense, BatchNormalization, Dropout, LSTM
from keras.models import Sequential
from keras import callbacks
from sklearn.metrics import precision_score, recall_score, confusion_matrix,␣
↪classification_report, accuracy_score, f1_score
```

[ ]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer
from sklearn.impute import KNNImputer
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score
from sklearn.metrics import confusion_matrix, roc_auc_score,␣
↪mean_absolute_error, mean_squared_error
#import xgboost as xgb
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
import scipy.stats as stats
```

[ ]:
```
combined = pd.read_csv('archive/Dataset.csv')
```

[ ]:
```
rows_to_drop = combined.loc[combined['Patient_ID'].apply(lambda x: len(str(x))␣
↪== 6)]
df_train = combined.drop(rows_to_drop.index)
df_train.to_csv('data_part1.csv', index=False)
```

```
rows_to_drop = combined.loc[combined['Patient_ID'].apply(lambda x: len(str(x)) !
↪= 6)]
df_test = combined.drop(rows_to_drop.index)
df_test.to_csv('data_part2.csv', index=False)
```

[ ]:
```
df_train = pd.read_csv('data_part1.csv')  # 45 col
df_test = pd.read_csv('data_part2.csv')
```

[ ]:
```
df_train.head(15)
df_train.columns
patients = list(df_test['Patient_ID'].unique())
len(patients)
```

[ ]: 20000

[ ]:
```
null_values = df_train.isnull().mean()*100
null_values = null_values.sort_values(ascending=False)
null_values

columns_drop={'Unnamed: 0','SBP','DBP','EtCO2','BaseExcess',␣
↪'HCO3','pH','PaCO2','Alkalinephos', 'Calcium','Magnesium',
'Phosphate','Potassium','PTT','Fibrinogen','Unit1','Unit2'}
df_train = df_train.assign(Unit=df_train['Unit1'] + df_train['Unit2'])
df_train_mod = df_train.drop(columns=columns_drop)
df_train_mod.columns
```

[ ]:
```
Index(['Hour', 'HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'FiO2', 'SaO2', 'AST',
       'BUN', 'Chloride', 'Creatinine', 'Bilirubin_direct', 'Glucose',
       'Lactate', 'Bilirubin_total', 'TroponinI', 'Hct', 'Hgb', 'WBC',
       'Platelets', 'Age', 'Gender', 'HospAdmTime', 'ICULOS', 'SepsisLabel',
       'Patient_ID', 'Unit'],
      dtype='object')
```

[ ]:
```
df_train_impute = df_train_mod.copy()
columns_impute = list(df_train_impute.columns)

grouped_by_patient = df_train_impute.groupby('Patient_ID')
df_train_impute = grouped_by_patient.apply(lambda x: x.bfill().ffill())
df_train_impute.head()

null_values = df_train_impute.isnull().mean()*100
null_values = null_values.sort_values(ascending=False)
null_values

null_col = ['TroponinI', 'Bilirubin_direct', 'AST', 'Bilirubin_total',␣
↪'Lactate', 'SaO2', 'FiO2',
            'Unit', 'Patient_ID']
```

```
df_train_impute = df_train_impute.drop(columns=null_col)
df_train_impute.columns
```

```
[ ]: Index(['Hour', 'HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'BUN', 'Chloride',
            'Creatinine', 'Glucose', 'Hct', 'Hgb', 'WBC', 'Platelets', 'Age',
            'Gender', 'HospAdmTime', 'ICULOS', 'SepsisLabel'],
           dtype='object')
```

```
[ ]: one_hot = pd.get_dummies(df_train_impute['Gender'])
df_train_impute = df_train_impute.join(one_hot)
df_train_impute = df_train_impute.drop('Gender', axis=1)

#df_train_impute = df_train_impute.drop(columns = ['col_yj','col_1.5','col_.
 ↪5','col_rec','col_log'])

df_train_impute.head()

columns_normalized = ['MAP', 'BUN', 'Creatinine', 'Glucose', 'WBC', 'Platelets'␣
 ↪]
for i in columns_normalized:
  df_train_impute[i] = np.log(df_train_impute[i]+1)

df_train_impute.head()

# standard normalization

scaler = StandardScaler()
df_train_impute[['HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'BUN', 'Chloride',
       'Creatinine', 'Glucose', 'Hct', 'Hgb', 'WBC', 'Platelets']] = scaler.
 ↪fit_transform(df_train_impute[['HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'BUN',␣
 ↪'Chloride',
       'Creatinine', 'Glucose', 'Hct', 'Hgb', 'WBC', 'Platelets']])
df_train_impute.head()
```

```
[ ]:    Hour        HR     O2Sat      Temp       MAP      Resp       BUN  Chloride  \
     0     0 -1.170030  0.865243 -1.548869 -0.397650 -0.419685  0.322965 -0.226541
     1     1 -1.170030  0.865243 -1.548869 -0.397650 -0.419685  0.322965 -0.226541
     2     2 -0.407913  0.865243 -1.548869 -3.187955 -0.326969  0.322965 -0.226541
     3     3 -0.701035  0.865243 -1.548869 -0.251970 -0.326969  0.322965 -0.226541
     4     4 -0.876908  0.865243 -1.548869 -0.251970 -0.883265  0.322965 -0.226541

        Creatinine   Glucose       Hct       Hgb       WBC  Platelets    Age  \
     0   -0.410796  0.854631 -0.311111 -0.652538  0.149678   1.140763  68.54
     1   -0.410796  0.854631 -0.311111 -0.652538  0.149678   1.140763  68.54
     2   -0.410796  0.854631 -0.311111 -0.652538  0.149678   1.140763  68.54
     3   -0.410796  0.854631 -0.311111 -0.652538  0.149678   1.140763  68.54
     4   -0.410796  0.854631 -0.311111 -0.652538  0.149678   1.140763  68.54
```

```
   HospAdmTime  ICULOS  SepsisLabel  0  1
0        -0.02       1            0  1  0
1        -0.02       2            0  1  0
2        -0.02       3            0  1  0
3        -0.02       4            0  1  0
4        -0.02       5            0  1  0
```

[ ]:
```
df_train_impute = df_train_impute.dropna()
null_values = df_train_impute.isnull().mean()*100
null_values

majority_class = df_train_impute[df_train_impute['SepsisLabel'] == 0]
minority_class = df_train_impute[df_train_impute['SepsisLabel'] == 1]
print('number of sepsis label 1 is {}'.format(len(minority_class)))
print('while number of sepsis label 0 is {}'.format(len(majority_class)))
majority_class_subset = majority_class.sample(n=2*len(minority_class))
df_train_impute = pd.concat([majority_class_subset, minority_class])
```

```
number of sepsis label 1 is 15284
while number of sepsis label 0 is 750935
```

[ ]:
```
def get_data_ready(df):
  columns_drop={'Unnamed: 0','SBP','DBP','EtCO2','BaseExcess',␣
↪'HCO3','pH','PaCO2','Alkalinephos', 'Calcium','Magnesium',
  'Phosphate','Potassium','PTT','Fibrinogen','Unit1','Unit2'}
  df = df.assign(Unit=df['Unit1'] + df['Unit2'])
  # dropping columns based on redundancy
  df = df.drop(columns=columns_drop)
  grouped_by_patient = df.groupby('Patient_ID')
  # imputing backfill and forward fill
  df = grouped_by_patient.apply(lambda x: x.bfill().ffill())
  # dropping all the columns with null values more than 25% and patient_id
  null_col = ['TroponinI', 'Bilirubin_direct', 'AST', 'Bilirubin_total',␣
↪'Lactate', 'SaO2', 'FiO2','Unit', 'Patient_ID']
  df = df.drop(columns=null_col)
  # gaussian transformation
  columns_normalized = ['MAP', 'BUN', 'Creatinine', 'Glucose', 'WBC',␣
↪'Platelets' ]
  for i in columns_normalized:
    df[i] = np.log(df[i]+1)
  # normailizing
  scaler = StandardScaler()
  df[['HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'BUN', 'Chloride',
       'Creatinine', 'Glucose', 'Hct', 'Hgb', 'WBC', 'Platelets']] = scaler.
↪fit_transform(df[['HR', 'O2Sat', 'Temp', 'MAP', 'Resp', 'BUN', 'Chloride',
       'Creatinine', 'Glucose', 'Hct', 'Hgb', 'WBC', 'Platelets']])
```

```
    # onehot encoding the gender
    one_hot = pd.get_dummies(df['Gender'])
    df = df.join(one_hot)
    df = df.drop('Gender', axis=1)
    df = df.dropna()
    return df
```

[ ]:
```
def evaluate_model(y_true,y_pred):
    accuracy = accuracy_score(y_true, y_pred)
    print("Accuracy:", accuracy)
    precision = precision_score(y_true, y_pred)
    print("Precision:", precision)
    recall = recall_score(y_true, y_pred)
    print("Recall:", recall)
    f1 = f1_score(y_true, y_pred)
    print("F1 Score:", f1)
    auc = roc_auc_score(y_true, y_pred)
    print("AUC-ROC:", auc)
    mae = mean_absolute_error(y_true, y_pred)
    print("Mean Absolute Error:", mae)
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    print("Root Mean Squared Error:", rmse)
    cm = confusion_matrix(y_true, y_pred)
    sns.heatmap(cm, annot=True, fmt='d')
    plt.show()
```

[ ]:
```
X = df_train_impute.drop('SepsisLabel', axis=1)
y = df_train_impute['SepsisLabel']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

[ ]:
```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(36681, 19)
(36681,)
(9171, 19)
(9171,)
```

[ ]:
```
# ANN
model = Sequential()

# layers
model.add(Dense(units = 256, kernel_initializer = 'uniform', activation =␣
 ↪'relu', input_dim = 19))
model.add(Dense(units = 128, kernel_initializer = 'uniform', activation =␣
 ↪'relu'))
```

```python
model.add(Dropout(0.25))
model.add(Dense(units = 64, kernel_initializer = 'uniform', activation =␣
 ↪'relu'))
model.add(Dropout(0.25))
model.add(Dense(units = 32, kernel_initializer = 'uniform', activation =␣
 ↪'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 16, kernel_initializer = 'uniform', activation =␣
 ↪'relu'))
model.add(Dropout(0.1))
model.add(Dense(units = 1, kernel_initializer = 'uniform', activation =␣
 ↪'sigmoid'))

# Compiling the ANN
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =␣
 ↪['accuracy']) # categorical_crossentropy binary_crossentropy

# Train the ANN
history = model.fit(X_train, y_train, batch_size = 64, epochs = 200,␣
 ↪validation_split=0.25)
```

```
Epoch 1/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5637 -
accuracy: 0.7378 - val_loss: 0.5372 - val_accuracy: 0.7380
Epoch 2/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5360 -
accuracy: 0.7526 - val_loss: 0.5330 - val_accuracy: 0.7432
Epoch 3/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5302 -
accuracy: 0.7541 - val_loss: 0.5305 - val_accuracy: 0.7429
Epoch 4/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5236 -
accuracy: 0.7583 - val_loss: 0.5226 - val_accuracy: 0.7599
Epoch 5/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5178 -
accuracy: 0.7605 - val_loss: 0.5134 - val_accuracy: 0.7592
Epoch 6/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5116 -
accuracy: 0.7610 - val_loss: 0.5126 - val_accuracy: 0.7561
Epoch 7/200
430/430 [==============================] - 1s 2ms/step - loss: 0.5076 -
accuracy: 0.7639 - val_loss: 0.5095 - val_accuracy: 0.7611
Epoch 8/200
430/430 [==============================] - 1s 3ms/step - loss: 0.5031 -
accuracy: 0.7662 - val_loss: 0.4997 - val_accuracy: 0.7628
Epoch 9/200
430/430 [==============================] - 2s 4ms/step - loss: 0.4999 -
```

```
accuracy: 0.7651 - val_loss: 0.4980 - val_accuracy: 0.7641
Epoch 10/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4928 -
accuracy: 0.7692 - val_loss: 0.4993 - val_accuracy: 0.7600
Epoch 11/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4916 -
accuracy: 0.7726 - val_loss: 0.4930 - val_accuracy: 0.7674
Epoch 12/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4872 -
accuracy: 0.7737 - val_loss: 0.4989 - val_accuracy: 0.7597
Epoch 13/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4811 -
accuracy: 0.7761 - val_loss: 0.4856 - val_accuracy: 0.7718
Epoch 14/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4782 -
accuracy: 0.7755 - val_loss: 0.4882 - val_accuracy: 0.7676
Epoch 15/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4714 -
accuracy: 0.7797 - val_loss: 0.4822 - val_accuracy: 0.7628
Epoch 16/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4692 -
accuracy: 0.7806 - val_loss: 0.4749 - val_accuracy: 0.7732
Epoch 17/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4641 -
accuracy: 0.7829 - val_loss: 0.4713 - val_accuracy: 0.7719
Epoch 18/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4606 -
accuracy: 0.7855 - val_loss: 0.4663 - val_accuracy: 0.7745
Epoch 19/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4537 -
accuracy: 0.7875 - val_loss: 0.4740 - val_accuracy: 0.7711
Epoch 20/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4523 -
accuracy: 0.7895 - val_loss: 0.4601 - val_accuracy: 0.7804
Epoch 21/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4467 -
accuracy: 0.7911 - val_loss: 0.4595 - val_accuracy: 0.7748
Epoch 22/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4389 -
accuracy: 0.7974 - val_loss: 0.4621 - val_accuracy: 0.7764
Epoch 23/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4361 -
accuracy: 0.7956 - val_loss: 0.4538 - val_accuracy: 0.7844
Epoch 24/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4338 -
accuracy: 0.7989 - val_loss: 0.4485 - val_accuracy: 0.7828
Epoch 25/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4274 -
```

```
accuracy: 0.7994 - val_loss: 0.4498 - val_accuracy: 0.7879
Epoch 26/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4237 -
accuracy: 0.8017 - val_loss: 0.4424 - val_accuracy: 0.7868
Epoch 27/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4167 -
accuracy: 0.8054 - val_loss: 0.4435 - val_accuracy: 0.7935
Epoch 28/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4138 -
accuracy: 0.8055 - val_loss: 0.4485 - val_accuracy: 0.7830
Epoch 29/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4105 -
accuracy: 0.8062 - val_loss: 0.4410 - val_accuracy: 0.7933
Epoch 30/200
430/430 [==============================] - 1s 2ms/step - loss: 0.4051 -
accuracy: 0.8104 - val_loss: 0.4233 - val_accuracy: 0.7994
Epoch 31/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3997 -
accuracy: 0.8132 - val_loss: 0.4264 - val_accuracy: 0.7977
Epoch 32/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3957 -
accuracy: 0.8182 - val_loss: 0.4338 - val_accuracy: 0.7946
Epoch 33/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3895 -
accuracy: 0.8192 - val_loss: 0.4299 - val_accuracy: 0.8043
Epoch 34/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3837 -
accuracy: 0.8237 - val_loss: 0.4264 - val_accuracy: 0.7930
Epoch 35/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3828 -
accuracy: 0.8216 - val_loss: 0.4216 - val_accuracy: 0.7986
Epoch 36/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3792 -
accuracy: 0.8259 - val_loss: 0.4202 - val_accuracy: 0.8006
Epoch 37/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3728 -
accuracy: 0.8277 - val_loss: 0.4240 - val_accuracy: 0.8082
Epoch 38/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3700 -
accuracy: 0.8314 - val_loss: 0.4162 - val_accuracy: 0.8077
Epoch 39/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3680 -
accuracy: 0.8308 - val_loss: 0.4128 - val_accuracy: 0.8118
Epoch 40/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3616 -
accuracy: 0.8323 - val_loss: 0.4006 - val_accuracy: 0.8159
Epoch 41/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3611 -
```

```
accuracy: 0.8354 - val_loss: 0.4003 - val_accuracy: 0.8142
Epoch 42/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3548 -
accuracy: 0.8399 - val_loss: 0.3974 - val_accuracy: 0.8186
Epoch 43/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3528 -
accuracy: 0.8391 - val_loss: 0.4127 - val_accuracy: 0.8109
Epoch 44/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3483 -
accuracy: 0.8400 - val_loss: 0.4027 - val_accuracy: 0.8144
Epoch 45/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3473 -
accuracy: 0.8432 - val_loss: 0.3934 - val_accuracy: 0.8155
Epoch 46/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3409 -
accuracy: 0.8447 - val_loss: 0.3940 - val_accuracy: 0.8187
Epoch 47/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3396 -
accuracy: 0.8481 - val_loss: 0.3978 - val_accuracy: 0.8234
Epoch 48/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3368 -
accuracy: 0.8463 - val_loss: 0.3862 - val_accuracy: 0.8243
Epoch 49/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3317 -
accuracy: 0.8506 - val_loss: 0.3795 - val_accuracy: 0.8282
Epoch 50/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3310 -
accuracy: 0.8525 - val_loss: 0.3819 - val_accuracy: 0.8238
Epoch 51/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3324 -
accuracy: 0.8517 - val_loss: 0.3970 - val_accuracy: 0.8193
Epoch 52/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3277 -
accuracy: 0.8521 - val_loss: 0.3945 - val_accuracy: 0.8274
Epoch 53/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3224 -
accuracy: 0.8562 - val_loss: 0.3837 - val_accuracy: 0.8262
Epoch 54/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3180 -
accuracy: 0.8550 - val_loss: 0.3862 - val_accuracy: 0.8335
Epoch 55/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3159 -
accuracy: 0.8593 - val_loss: 0.3980 - val_accuracy: 0.8313
Epoch 56/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3127 -
accuracy: 0.8623 - val_loss: 0.3942 - val_accuracy: 0.8289
Epoch 57/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3157 -
```

```
accuracy: 0.8609 - val_loss: 0.3728 - val_accuracy: 0.8369
Epoch 58/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3108 -
accuracy: 0.8628 - val_loss: 0.3793 - val_accuracy: 0.8346
Epoch 59/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3118 -
accuracy: 0.8613 - val_loss: 0.3746 - val_accuracy: 0.8365
Epoch 60/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3049 -
accuracy: 0.8675 - val_loss: 0.3936 - val_accuracy: 0.8367
Epoch 61/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3074 -
accuracy: 0.8631 - val_loss: 0.3958 - val_accuracy: 0.8290
Epoch 62/200
430/430 [==============================] - 1s 2ms/step - loss: 0.3043 -
accuracy: 0.8642 - val_loss: 0.3748 - val_accuracy: 0.8385
Epoch 63/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2988 -
accuracy: 0.8672 - val_loss: 0.3790 - val_accuracy: 0.8373
Epoch 64/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2921 -
accuracy: 0.8690 - val_loss: 0.3846 - val_accuracy: 0.8376
Epoch 65/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2916 -
accuracy: 0.8718 - val_loss: 0.3892 - val_accuracy: 0.8393
Epoch 66/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2900 -
accuracy: 0.8737 - val_loss: 0.3747 - val_accuracy: 0.8407
Epoch 67/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2946 -
accuracy: 0.8695 - val_loss: 0.3692 - val_accuracy: 0.8429
Epoch 68/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2907 -
accuracy: 0.8720 - val_loss: 0.3636 - val_accuracy: 0.8452
Epoch 69/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2877 -
accuracy: 0.8745 - val_loss: 0.3727 - val_accuracy: 0.8469
Epoch 70/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2877 -
accuracy: 0.8727 - val_loss: 0.3750 - val_accuracy: 0.8448
Epoch 71/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2819 -
accuracy: 0.8767 - val_loss: 0.3757 - val_accuracy: 0.8432
Epoch 72/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2831 -
accuracy: 0.8755 - val_loss: 0.3729 - val_accuracy: 0.8441
Epoch 73/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2777 -
```

```
accuracy: 0.8773 - val_loss: 0.3623 - val_accuracy: 0.8431
Epoch 74/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2783 -
accuracy: 0.8790 - val_loss: 0.3658 - val_accuracy: 0.8466
Epoch 75/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2806 -
accuracy: 0.8760 - val_loss: 0.3669 - val_accuracy: 0.8423
Epoch 76/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2738 -
accuracy: 0.8792 - val_loss: 0.3756 - val_accuracy: 0.8445
Epoch 77/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2765 -
accuracy: 0.8767 - val_loss: 0.3764 - val_accuracy: 0.8409
Epoch 78/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2775 -
accuracy: 0.8791 - val_loss: 0.3667 - val_accuracy: 0.8458
Epoch 79/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2718 -
accuracy: 0.8814 - val_loss: 0.3663 - val_accuracy: 0.8455
Epoch 80/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2698 -
accuracy: 0.8822 - val_loss: 0.3566 - val_accuracy: 0.8500
Epoch 81/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2762 -
accuracy: 0.8781 - val_loss: 0.3637 - val_accuracy: 0.8495
Epoch 82/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2706 -
accuracy: 0.8815 - val_loss: 0.3747 - val_accuracy: 0.8458
Epoch 83/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2686 -
accuracy: 0.8823 - val_loss: 0.3646 - val_accuracy: 0.8495
Epoch 84/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2634 -
accuracy: 0.8851 - val_loss: 0.3763 - val_accuracy: 0.8481
Epoch 85/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2635 -
accuracy: 0.8849 - val_loss: 0.3539 - val_accuracy: 0.8539
Epoch 86/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2641 -
accuracy: 0.8841 - val_loss: 0.3498 - val_accuracy: 0.8558
Epoch 87/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2555 -
accuracy: 0.8874 - val_loss: 0.3865 - val_accuracy: 0.8468
Epoch 88/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2606 -
accuracy: 0.8865 - val_loss: 0.4061 - val_accuracy: 0.8477
Epoch 89/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2614 -
```

```
accuracy: 0.8842 - val_loss: 0.3721 - val_accuracy: 0.8547
Epoch 90/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2576 -
accuracy: 0.8856 - val_loss: 0.3568 - val_accuracy: 0.8581
Epoch 91/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2522 -
accuracy: 0.8892 - val_loss: 0.3826 - val_accuracy: 0.8539
Epoch 92/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2545 -
accuracy: 0.8908 - val_loss: 0.3615 - val_accuracy: 0.8515
Epoch 93/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2551 -
accuracy: 0.8886 - val_loss: 0.3691 - val_accuracy: 0.8496
Epoch 94/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2525 -
accuracy: 0.8890 - val_loss: 0.3720 - val_accuracy: 0.8543
Epoch 95/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2481 -
accuracy: 0.8914 - val_loss: 0.3742 - val_accuracy: 0.8544
Epoch 96/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2509 -
accuracy: 0.8909 - val_loss: 0.3588 - val_accuracy: 0.8603
Epoch 97/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2483 -
accuracy: 0.8907 - val_loss: 0.3665 - val_accuracy: 0.8491
Epoch 98/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2522 -
accuracy: 0.8909 - val_loss: 0.3585 - val_accuracy: 0.8577
Epoch 99/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2476 -
accuracy: 0.8929 - val_loss: 0.3552 - val_accuracy: 0.8579
Epoch 100/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2488 -
accuracy: 0.8926 - val_loss: 0.3445 - val_accuracy: 0.8589
Epoch 101/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2471 -
accuracy: 0.8924 - val_loss: 0.3545 - val_accuracy: 0.8568
Epoch 102/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2427 -
accuracy: 0.8947 - val_loss: 0.3696 - val_accuracy: 0.8560
Epoch 103/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2465 -
accuracy: 0.8923 - val_loss: 0.3755 - val_accuracy: 0.8539
Epoch 104/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2407 -
accuracy: 0.8945 - val_loss: 0.3761 - val_accuracy: 0.8532
Epoch 105/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2366 -
```

```
accuracy: 0.8955 - val_loss: 0.3623 - val_accuracy: 0.8624
Epoch 106/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2404 -
accuracy: 0.8960 - val_loss: 0.3618 - val_accuracy: 0.8604
Epoch 107/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2442 -
accuracy: 0.8943 - val_loss: 0.3628 - val_accuracy: 0.8616
Epoch 108/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2454 -
accuracy: 0.8935 - val_loss: 0.3501 - val_accuracy: 0.8582
Epoch 109/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2437 -
accuracy: 0.8946 - val_loss: 0.3662 - val_accuracy: 0.8581
Epoch 110/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2417 -
accuracy: 0.8952 - val_loss: 0.3621 - val_accuracy: 0.8590
Epoch 111/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2371 -
accuracy: 0.8985 - val_loss: 0.3752 - val_accuracy: 0.8673
Epoch 112/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2388 -
accuracy: 0.8968 - val_loss: 0.3483 - val_accuracy: 0.8642
Epoch 113/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2341 -
accuracy: 0.8984 - val_loss: 0.3601 - val_accuracy: 0.8600
Epoch 114/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2378 -
accuracy: 0.8976 - val_loss: 0.3613 - val_accuracy: 0.8632
Epoch 115/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2328 -
accuracy: 0.8998 - val_loss: 0.3718 - val_accuracy: 0.8603
Epoch 116/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2376 -
accuracy: 0.8971 - val_loss: 0.3830 - val_accuracy: 0.8622
Epoch 117/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2394 -
accuracy: 0.8962 - val_loss: 0.3691 - val_accuracy: 0.8569
Epoch 118/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2340 -
accuracy: 0.8996 - val_loss: 0.3761 - val_accuracy: 0.8589
Epoch 119/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2307 -
accuracy: 0.9003 - val_loss: 0.3485 - val_accuracy: 0.8648
Epoch 120/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2324 -
accuracy: 0.9007 - val_loss: 0.3688 - val_accuracy: 0.8626
Epoch 121/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2307 -
```

```
accuracy: 0.8988 - val_loss: 0.3497 - val_accuracy: 0.8665
Epoch 122/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2297 -
accuracy: 0.9003 - val_loss: 0.3672 - val_accuracy: 0.8591
Epoch 123/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2300 -
accuracy: 0.9016 - val_loss: 0.3503 - val_accuracy: 0.8634
Epoch 124/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2248 -
accuracy: 0.9020 - val_loss: 0.3588 - val_accuracy: 0.8624
Epoch 125/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2278 -
accuracy: 0.9023 - val_loss: 0.3490 - val_accuracy: 0.8645
Epoch 126/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2290 -
accuracy: 0.9000 - val_loss: 0.3675 - val_accuracy: 0.8615
Epoch 127/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2292 -
accuracy: 0.8994 - val_loss: 0.3551 - val_accuracy: 0.8661
Epoch 128/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2238 -
accuracy: 0.9027 - val_loss: 0.3558 - val_accuracy: 0.8616
Epoch 129/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2264 -
accuracy: 0.9027 - val_loss: 0.3624 - val_accuracy: 0.8639
Epoch 130/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2291 -
accuracy: 0.9006 - val_loss: 0.3951 - val_accuracy: 0.8541
Epoch 131/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2226 -
accuracy: 0.9039 - val_loss: 0.3670 - val_accuracy: 0.8581
Epoch 132/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2237 -
accuracy: 0.9037 - val_loss: 0.3564 - val_accuracy: 0.8671
Epoch 133/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2262 -
accuracy: 0.9020 - val_loss: 0.3527 - val_accuracy: 0.8664
Epoch 134/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2240 -
accuracy: 0.9005 - val_loss: 0.3431 - val_accuracy: 0.8652
Epoch 135/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2201 -
accuracy: 0.9065 - val_loss: 0.3689 - val_accuracy: 0.8661
Epoch 136/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2212 -
accuracy: 0.9062 - val_loss: 0.3475 - val_accuracy: 0.8621
Epoch 137/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2225 -
```

```
accuracy: 0.9033 - val_loss: 0.3582 - val_accuracy: 0.8695
Epoch 138/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2211 -
accuracy: 0.9031 - val_loss: 0.3579 - val_accuracy: 0.8684
Epoch 139/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2153 -
accuracy: 0.9087 - val_loss: 0.3544 - val_accuracy: 0.8673
Epoch 140/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2198 -
accuracy: 0.9055 - val_loss: 0.3857 - val_accuracy: 0.8519
Epoch 141/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2217 -
accuracy: 0.9036 - val_loss: 0.3556 - val_accuracy: 0.8722
Epoch 142/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2097 -
accuracy: 0.9078 - val_loss: 0.3652 - val_accuracy: 0.8630
Epoch 143/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2181 -
accuracy: 0.9056 - val_loss: 0.3482 - val_accuracy: 0.8742
Epoch 144/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2123 -
accuracy: 0.9071 - val_loss: 0.3635 - val_accuracy: 0.8700
Epoch 145/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2119 -
accuracy: 0.9094 - val_loss: 0.3725 - val_accuracy: 0.8674
Epoch 146/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2149 -
accuracy: 0.9071 - val_loss: 0.3577 - val_accuracy: 0.8663
Epoch 147/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2125 -
accuracy: 0.9085 - val_loss: 0.3564 - val_accuracy: 0.8669
Epoch 148/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2089 -
accuracy: 0.9107 - val_loss: 0.3682 - val_accuracy: 0.8681
Epoch 149/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2116 -
accuracy: 0.9085 - val_loss: 0.3891 - val_accuracy: 0.8674
Epoch 150/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2070 -
accuracy: 0.9104 - val_loss: 0.3753 - val_accuracy: 0.8693
Epoch 151/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2179 -
accuracy: 0.9043 - val_loss: 0.3621 - val_accuracy: 0.8662
Epoch 152/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2126 -
accuracy: 0.9083 - val_loss: 0.3693 - val_accuracy: 0.8710
Epoch 153/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2080 -
```

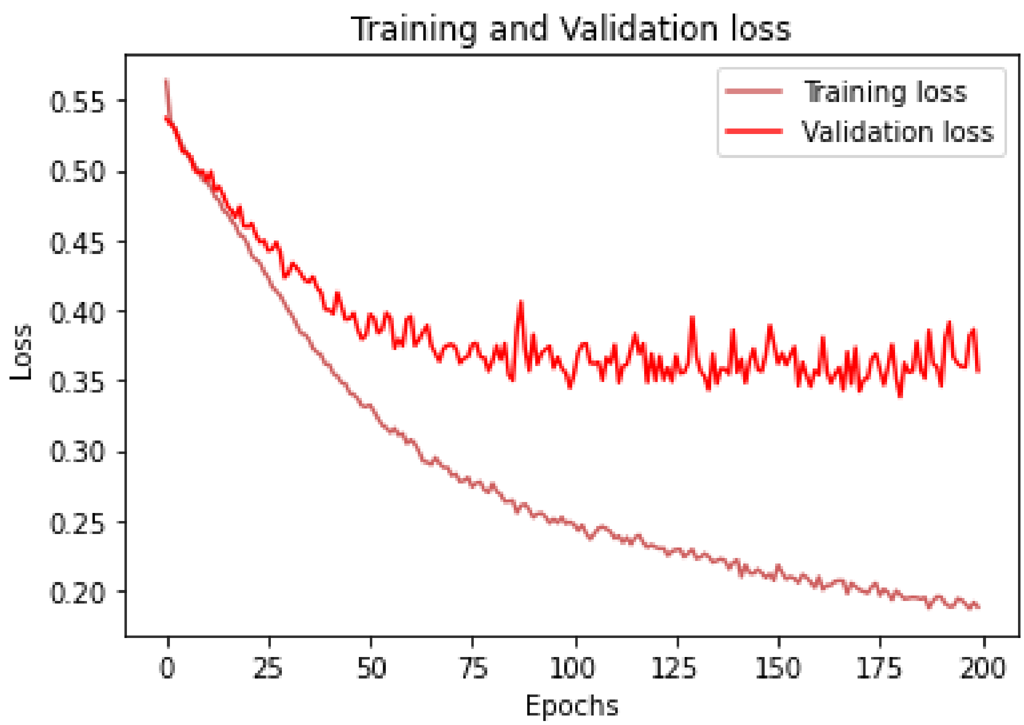
```
accuracy: 0.9093 - val_loss: 0.3611 - val_accuracy: 0.8721
Epoch 154/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2096 -
accuracy: 0.9084 - val_loss: 0.3679 - val_accuracy: 0.8699
Epoch 155/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2085 -
accuracy: 0.9092 - val_loss: 0.3736 - val_accuracy: 0.8717
Epoch 156/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2053 -
accuracy: 0.9113 - val_loss: 0.3460 - val_accuracy: 0.8717
Epoch 157/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2111 -
accuracy: 0.9087 - val_loss: 0.3627 - val_accuracy: 0.8686
Epoch 158/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2092 -
accuracy: 0.9113 - val_loss: 0.3532 - val_accuracy: 0.8711
Epoch 159/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2058 -
accuracy: 0.9107 - val_loss: 0.3456 - val_accuracy: 0.8729
Epoch 160/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2019 -
accuracy: 0.9141 - val_loss: 0.3570 - val_accuracy: 0.8775
Epoch 161/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2094 -
accuracy: 0.9091 - val_loss: 0.3540 - val_accuracy: 0.8743
Epoch 162/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2004 -
accuracy: 0.9136 - val_loss: 0.3806 - val_accuracy: 0.8684
Epoch 163/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2034 -
accuracy: 0.9123 - val_loss: 0.3560 - val_accuracy: 0.8659
Epoch 164/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2036 -
accuracy: 0.9120 - val_loss: 0.3479 - val_accuracy: 0.8714
Epoch 165/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2063 -
accuracy: 0.9125 - val_loss: 0.3544 - val_accuracy: 0.8696
Epoch 166/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2069 -
accuracy: 0.9112 - val_loss: 0.3591 - val_accuracy: 0.8680
Epoch 167/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2066 -
accuracy: 0.9114 - val_loss: 0.3432 - val_accuracy: 0.8782
Epoch 168/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1976 -
accuracy: 0.9153 - val_loss: 0.3702 - val_accuracy: 0.8749
Epoch 169/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2050 -
```

```
accuracy: 0.9120 - val_loss: 0.3473 - val_accuracy: 0.8696
Epoch 170/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2028 -
accuracy: 0.9126 - val_loss: 0.3733 - val_accuracy: 0.8721
Epoch 171/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2003 -
accuracy: 0.9136 - val_loss: 0.3421 - val_accuracy: 0.8720
Epoch 172/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1992 -
accuracy: 0.9142 - val_loss: 0.3503 - val_accuracy: 0.8777
Epoch 173/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1980 -
accuracy: 0.9146 - val_loss: 0.3515 - val_accuracy: 0.8730
Epoch 174/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2027 -
accuracy: 0.9133 - val_loss: 0.3655 - val_accuracy: 0.8718
Epoch 175/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2049 -
accuracy: 0.9118 - val_loss: 0.3688 - val_accuracy: 0.8683
Epoch 176/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1962 -
accuracy: 0.9167 - val_loss: 0.3578 - val_accuracy: 0.8755
Epoch 177/200
430/430 [==============================] - 1s 2ms/step - loss: 0.2007 -
accuracy: 0.9126 - val_loss: 0.3468 - val_accuracy: 0.8754
Epoch 178/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1983 -
accuracy: 0.9169 - val_loss: 0.3597 - val_accuracy: 0.8737
Epoch 179/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1932 -
accuracy: 0.9161 - val_loss: 0.3754 - val_accuracy: 0.8712
Epoch 180/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1996 -
accuracy: 0.9149 - val_loss: 0.3515 - val_accuracy: 0.8688
Epoch 181/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1968 -
accuracy: 0.9171 - val_loss: 0.3381 - val_accuracy: 0.8735
Epoch 182/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1938 -
accuracy: 0.9156 - val_loss: 0.3624 - val_accuracy: 0.8678
Epoch 183/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1943 -
accuracy: 0.9180 - val_loss: 0.3556 - val_accuracy: 0.8761
Epoch 184/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1949 -
accuracy: 0.9155 - val_loss: 0.3566 - val_accuracy: 0.8708
Epoch 185/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1946 -
```

```
accuracy: 0.9178 - val_loss: 0.3776 - val_accuracy: 0.8726
Epoch 186/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1938 -
accuracy: 0.9184 - val_loss: 0.3574 - val_accuracy: 0.8708
Epoch 187/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1948 -
accuracy: 0.9177 - val_loss: 0.3514 - val_accuracy: 0.8761
Epoch 188/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1874 -
accuracy: 0.9207 - val_loss: 0.3854 - val_accuracy: 0.8729
Epoch 189/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1923 -
accuracy: 0.9173 - val_loss: 0.3620 - val_accuracy: 0.8746
Epoch 190/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1954 -
accuracy: 0.9166 - val_loss: 0.3593 - val_accuracy: 0.8760
Epoch 191/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1951 -
accuracy: 0.9162 - val_loss: 0.3459 - val_accuracy: 0.8781
Epoch 192/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1904 -
accuracy: 0.9189 - val_loss: 0.3800 - val_accuracy: 0.8748
Epoch 193/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1883 -
accuracy: 0.9189 - val_loss: 0.3914 - val_accuracy: 0.8730
Epoch 194/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1886 -
accuracy: 0.9189 - val_loss: 0.3664 - val_accuracy: 0.8755
Epoch 195/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1939 -
accuracy: 0.9168 - val_loss: 0.3620 - val_accuracy: 0.8718
Epoch 196/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1929 -
accuracy: 0.9181 - val_loss: 0.3596 - val_accuracy: 0.8778
Epoch 197/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1900 -
accuracy: 0.9180 - val_loss: 0.3594 - val_accuracy: 0.8743
Epoch 198/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1868 -
accuracy: 0.9205 - val_loss: 0.3821 - val_accuracy: 0.8767
Epoch 199/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1913 -
accuracy: 0.9191 - val_loss: 0.3862 - val_accuracy: 0.8697
Epoch 200/200
430/430 [==============================] - 1s 2ms/step - loss: 0.1880 -
accuracy: 0.9211 - val_loss: 0.3568 - val_accuracy: 0.8790
```

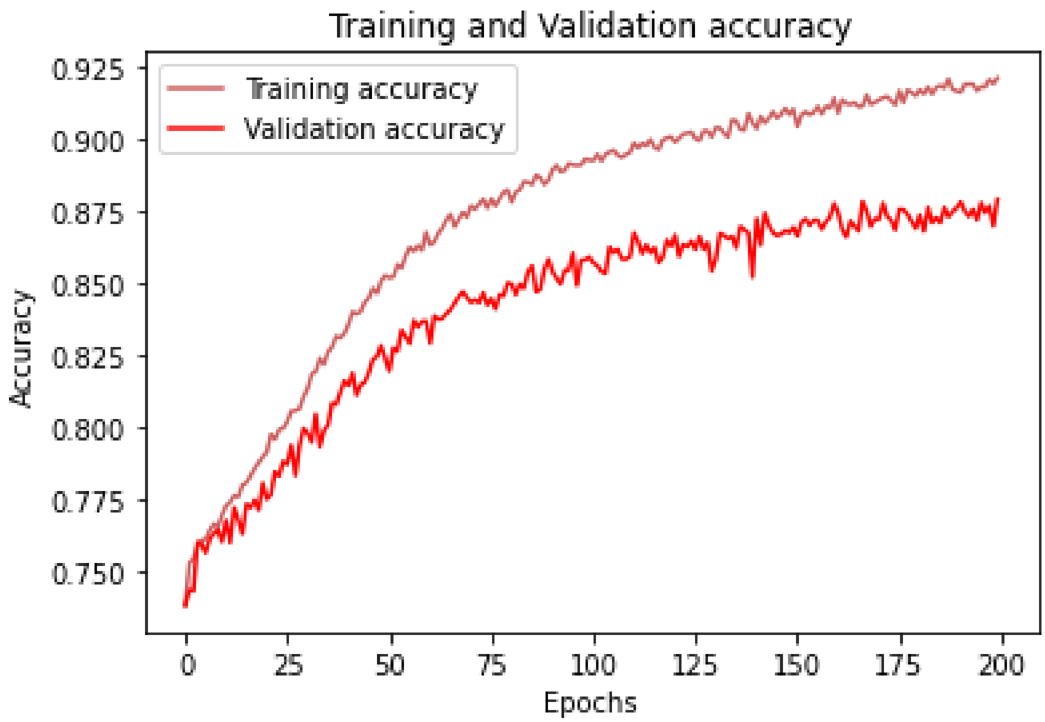
[ ]:
```
val_accuracy = np.mean(history.history['val_accuracy'])
print("\n%s: %.2f%%" % ('val_accuracy is', val_accuracy*100))

history_df = pd.DataFrame(history.history)

plt.plot(history_df.loc[:, ['loss']], "#CD5C5C", label='Training loss')
plt.plot(history_df.loc[:, ['val_loss']],"#FF0000", label='Validation loss')
plt.title('Training and Validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(loc="best")

plt.show()
```

```
val_accuracy is: 84.17%
```



[ ]:
```
history_df = pd.DataFrame(history.history)

plt.plot(history_df.loc[:, ['accuracy']], "#CD5C5C", label='Training accuracy')
plt.plot(history_df.loc[:, ['val_accuracy']],"#FF0000", label='Validation␣
↪accuracy')

plt.title('Training and Validation accuracy')
```

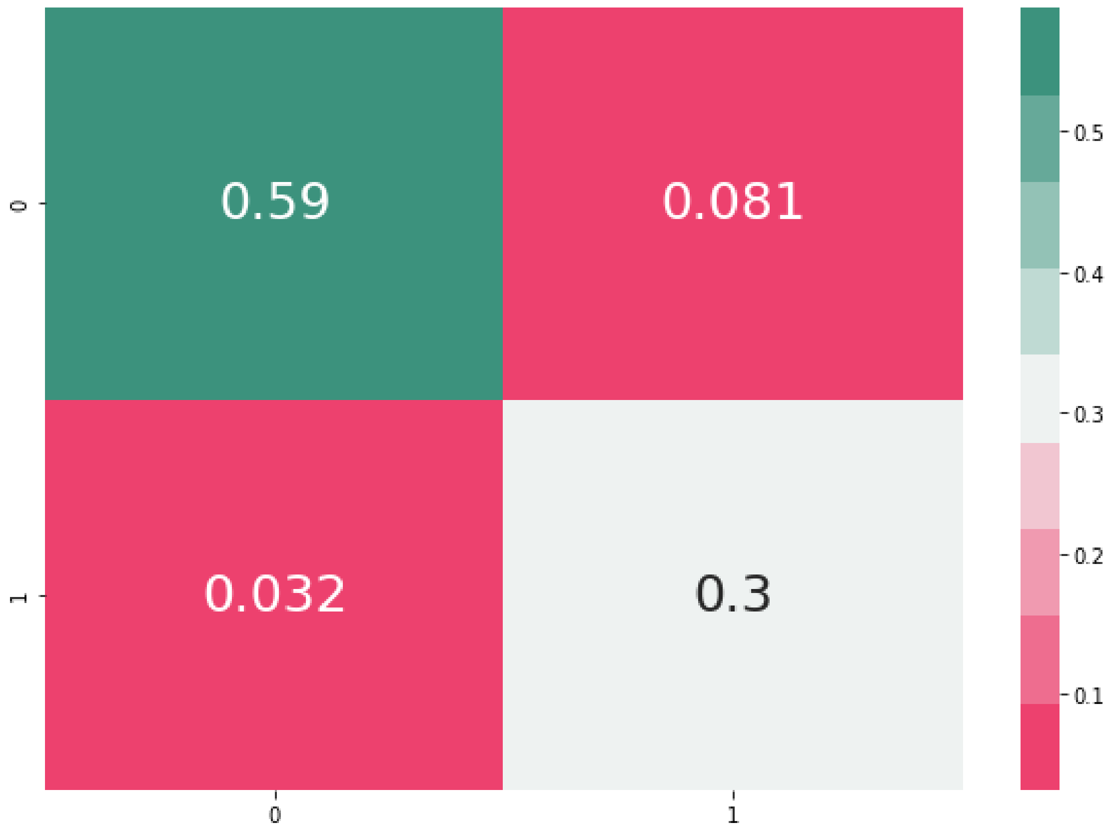
```
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



[ ]:
```
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.4)
np.set_printoptions()

# Getting the confusion matrix
cmap1 = sns.diverging_palette(2, 165, s=80, l=55, n=9)
plt.subplots(figsize=(10,7))
cf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(cf_matrix/np.sum(cf_matrix), cmap = cmap1, annot = True, annot_kws␣
↪= {'size':25})
```

```
287/287 [==============================] - 0s 746us/step
```

[ ]: <Axes: >



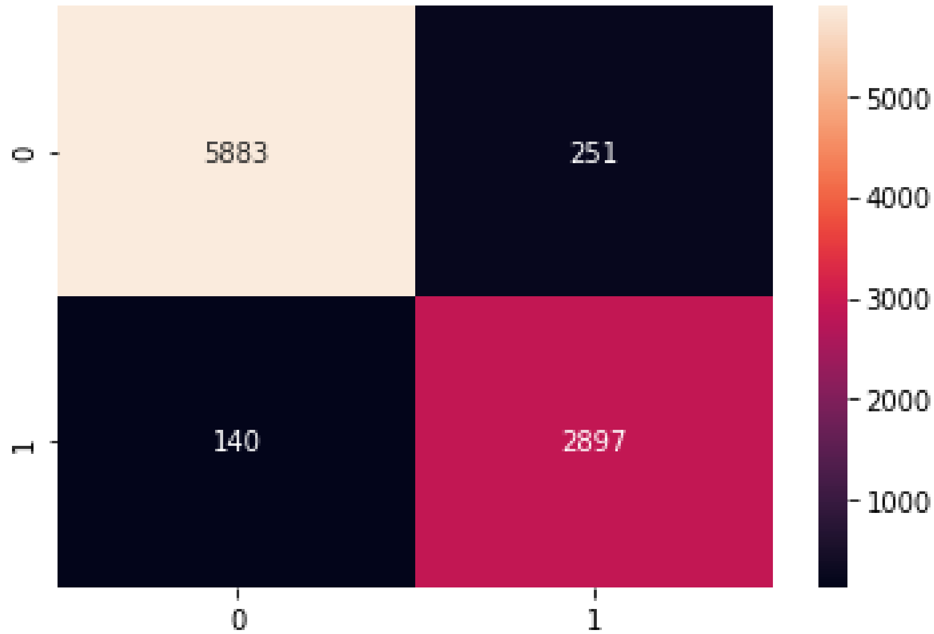
[ ]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      0.88      0.91      6134
           1       0.79      0.90      0.84      3037

    accuracy                           0.89      9171
   macro avg       0.87      0.89      0.88      9171
weighted avg       0.89      0.89      0.89      9171
```

[ ]:
```
model = RandomForestClassifier(n_estimators=100, random_state=0)
hist=model.fit(X_train, y_train)
rcf_predictions = model.predict(X_test)

evaluate_model(y_test,rcf_predictions)
```

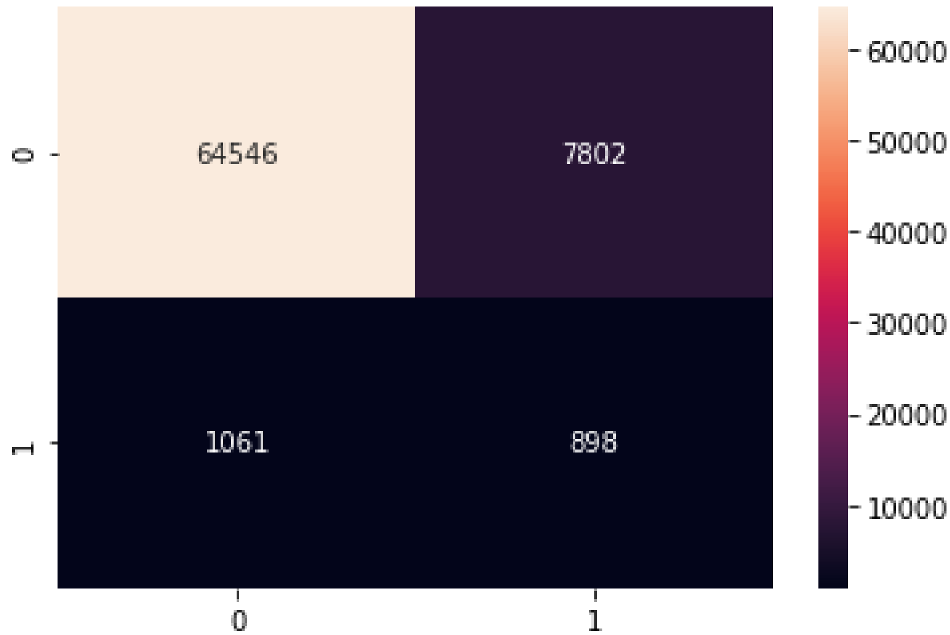
```
Accuracy: 0.9573656089848436
Precision: 0.920266836086404
Recall: 0.9539018768521568
F1 Score: 0.9367825383993533
```

```
AUC-ROC: 0.9564912057883216
Mean Absolute Error: 0.042634391015156474
Root Mean Squared Error: 0.20648097010416352
```



[ ]:
```
df = get_data_ready(df_test)
X = df.drop('SepsisLabel', axis=1)
y = df['SepsisLabel']
rcf_predictions = model.predict(X)
evaluate_model(y,rcf_predictions)
```

```
Accuracy: 0.8807245616159985
Precision: 0.1032183908045977
Recall: 0.45839714139867277
F1 Score: 0.16849610657660194
AUC-ROC: 0.6752786281991981
Mean Absolute Error: 0.11927543838400151
Root Mean Squared Error: 0.34536276345894834
```



[ ]:
```
print(classification_report(y, rcf_predictions))
```

```
              precision    recall  f1-score   support

           0       0.98      0.89      0.94     72348
           1       0.10      0.46      0.17      1959

    accuracy                           0.88     74307
   macro avg       0.54      0.68      0.55     74307
weighted avg       0.96      0.88      0.92     74307
```