PS6

January 15, 2024

[1]:

```python
from shap.datasets import adult

X, y = adult()

print(X)
print(y)
```

```
        Age  Workclass  Education-Num  Marital Status  Occupation  \
0      39.0          7           13.0               4           1
1      50.0          6           13.0               2           4
2      38.0          4            9.0               0           6
3      53.0          4            7.0               2           6
4      28.0          4           13.0               2          10
…     …        …            …             …         …
32556  27.0          4           12.0               2          13
32557  40.0          4            9.0               2           7
32558  58.0          4            9.0               6           1
32559  22.0          4            9.0               4           1
32560  52.0          5            9.0               2           4

       Relationship  Race  Sex  Capital Gain  Capital Loss  Hours per week  \
0                 0     4    1        2174.0           0.0            40.0
1                 4     4    1           0.0           0.0            13.0
2                 0     4    1           0.0           0.0            40.0
3                 4     2    1           0.0           0.0            40.0
4                 5     2    0           0.0           0.0            40.0
…             …   …  …           …           …             …
32556             5     4    0           0.0           0.0            38.0
32557             4     4    1           0.0           0.0            40.0
32558             1     4    0           0.0           0.0            40.0
32559             3     4    1           0.0           0.0            20.0
32560             5     4    0       15024.0           0.0            40.0

       Country
0           39
1           39
2           39
3           39
```

```
4             5
…          …
32556       39
32557       39
32558       39
32559       39
32560       39

[32561 rows x 12 columns]
[False False False … False False  True]
```

[2]:
```python
numerical_columns = ['Age','Education-Num','Capital Gain','Capital Loss','Hours per week']
categorical_columns = ['Workclass','Marital Status','Occupation','Relationship','Race','Sex','Country']
```

0.1 Conversion of categorical data

[3]:
```python
import pandas as pd # for one-hot encoding

from sklearn.preprocessing import StandardScaler # for normalization
```

[4]:
```python
# Normalization of numerical data
for column in numerical_columns:
    scaler = StandardScaler()
    X[column] = scaler.fit_transform(X[column].values.reshape(-1,1))

print(X)
```

```
            Age  Workclass  Education-Num  Marital Status  Occupation  \
0      0.030671          7       1.134739               4           1
1      0.837109          6       1.134739               2           4
2     -0.042642          4      -0.420060               0           6
3      1.057047          4      -1.197459               2           6
4     -0.775768          4       1.134739               2          10
…           …        …            …             …           …
32556 -0.849080          4       0.746039               2          13
32557  0.103983          4      -0.420060               2           7
32558  1.423610          4      -0.420060               6           1
32559 -1.215643          4      -0.420060               4           1
32560  0.983734          5      -0.420060               2           4

       Relationship  Race  Sex  Capital Gain  Capital Loss  Hours per week  \
0                 0     4    1      0.148453      -0.21666       -0.035429
1                 4     4    1     -0.145920      -0.21666       -2.222153
2                 0     4    1     -0.145920      -0.21666       -0.035429
3                 4     2    1     -0.145920      -0.21666       -0.035429
4                 5     2    0     -0.145920      -0.21666       -0.035429
```

```
…             …   …  …           …           …             …
32556             5     4    0     -0.145920      -0.21666       -0.197409
32557             4     4    1     -0.145920      -0.21666       -0.035429
32558             1     4    0     -0.145920      -0.21666       -0.035429
32559             3     4    1     -0.145920      -0.21666       -1.655225
32560             5     4    0      1.888424      -0.21666       -0.035429

       Country
0            39
1            39
2            39
3            39
4             5
…        …
32556        39
32557        39
32558        39
32559        39
32560        39

[32561 rows x 12 columns]
```

[5]:

```python
# Data type change of categorical data
for column in categorical_columns:
    X[column] = X[column].astype('category')

print(X)
```

```
            Age Workclass  Education-Num Marital Status Occupation  \
0      0.030671         7       1.134739              4          1
1      0.837109         6       1.134739              2          4
2     -0.042642         4      -0.420060              0          6
3      1.057047         4      -1.197459              2          6
4     -0.775768         4       1.134739              2         10
…         …       …            …            …        …
32556 -0.849080         4       0.746039              2         13
32557  0.103983         4      -0.420060              2          7
32558  1.423610         4      -0.420060              6          1
32559 -1.215643         4      -0.420060              4          1
32560  0.983734         5      -0.420060              2          4

      Relationship Race Sex  Capital Gain  Capital Loss  Hours per week  \
0                0    4   1      0.148453      -0.21666       -0.035429
1                4    4   1     -0.145920      -0.21666       -2.222153
2                0    4   1     -0.145920      -0.21666       -0.035429
3                4    2   1     -0.145920      -0.21666       -0.035429
4                5    2   0     -0.145920      -0.21666       -0.035429
…            …  …  ..           …           …             …
```

```
32556          5    4    0    -0.145920    -0.21666    -0.197409
32557          4    4    1    -0.145920    -0.21666    -0.035429
32558          1    4    0    -0.145920    -0.21666    -0.035429
32559          3    4    1    -0.145920    -0.21666    -1.655225
32560          5    4    0     1.888424    -0.21666    -0.035429

      Country
0          39
1          39
2          39
3          39
4           5
…        …
32556      39
32557      39
32558      39
32559      39
32560      39

[32561 rows x 12 columns]
```

[6]:
```
# One-hot encoding of categorical data
X = pd.get_dummies(X)

# Conversion of data frame to numpy
X = X.values

# Converision: {False, True} --> {0., 1.}
y = y.astype(float)
```

[7]:
```
print(X.shape)
print(y.shape)
print(y)
```

```
(32561, 91)
(32561,)
[0. 0. 0. … 0. 0. 1.]
```

0.2 train-val-test split

[8]:
```
from sklearn.model_selection import train_test_split

X_,X_test,y_,y_test = train_test_split(X,y,test_size=1/10,stratify=y)
X_train,X_val,y_train,y_val = train_test_split(X_,y_,test_size=1/9,stratify=y_)

print(X_train.shape)
print(X_val.shape)
print(X_test.shape)
```

```
(26048, 91)
(3256, 91)
(3257, 91)
```

0.3 Cross validation

[9]:

```
import numpy as np
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

X1  = np.array([10, 20, 30, 40, 50])
y1  = np.array([60, 70, 80, 90, 100])

kfold = KFold(n_splits=4,shuffle=True)

X1_,X1_test,y1_,y1_test = train_test_split(X1,y1,test_size=1/5)

print(kfold.get_n_splits())
print(X1_)
print(y1_)
```

```
4
[40 50 20 10]
[ 90 100  70  60]
```

[10]:

```
for train, val in kfold.split(X1_,y1_):
    print("Train indices: ",train)
    X1_train = X1_[train]
    print("Train datasets: ",X1_train)
```

```
Train indices:  [0 1 2]
Train datasets:  [40 50 20]
Train indices:  [1 2 3]
Train datasets:  [50 20 10]
Train indices:  [0 1 3]
Train datasets:  [40 50 10]
Train indices:  [0 2 3]
Train datasets:  [40 20 10]
```

[11]:

```
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

#kfold = KFold(n_splits=4, shuffle=True)
kfold = KFold(n_splits=4)

X_,X_test,y_,y_test = train_test_split(X,y,test_size=1/10,stratify=y)

aaa = kfold.split(X_,y_)
```

```python
for train, val in aaa:
    print(train)
    #print(val)
    X_train = X[train]
    print(X_train[0])
   # print(X_train.shape)
```

```
[ 7326  7327  7328 … 29301 29302 29303]
[-0.11595462  0.7460392  -0.14592049 -0.21665953 -0.19740899  0.
  0.          0.          0.          0.          0.          0.
  1.          0.          1.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          1.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          1.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  0.        ]
[    0     1     2 … 29301 29302 29303]
[ 0.03067056  1.1347387   0.14845291 -0.21665953 -0.03542945  0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          1.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  0.        ]
[    0     1     2 … 29301 29302 29303]
[ 0.03067056  1.1347387   0.14845291 -0.21665953 -0.03542945  0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          1.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  0.        ]
[    0     1     2 … 21975 21976 21977]
[ 0.03067056  1.1347387   0.14845291 -0.21665953 -0.03542945  0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          1.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  1.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          1.          0.
  0.        ]
```

[12]:

```
def scheduler(epoch, lr):
    if epoch in [20,40,60]:
        lr = 0.1*lr
    else:
        lr = lr
    return lr
```

[13]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
from tensorflow.keras.initializers import HeNormal
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import ReLU
from tensorflow.keras.layers import Dropout
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import LearningRateScheduler
from sklearn.model_selection import KFold

kfold = KFold(n_splits=4, shuffle=True)

X_,X_test,y_,y_test = train_test_split(X,y,test_size=1/10,stratify=y)

for train, val in kfold.split(X_,y_):

    # Train and val datasets
    X_train, X_val = X[train], X[val]
    y_train, y_val = y[train], y[val]
    #print(X_train.shape)
    #print(X_val.shape)

    # Construct a model
    init = HeNormal()
    model = Sequential()
    model.add(Dense(128,kernel_regularizer=l2(0.01),
              bias_regularizer=l2(0.01),
              kernel_initializer=init))
    model.add(BatchNormalization())
    model.add(ReLU())
    model.add(Dropout(0.5))
    model.add(Dense(1,activation='sigmoid'))

    # Compile
    opt = Adam(learning_rate=0.01,beta_1 = 0.9,beta_2 = 0.999)
    model.compile(optimizer=opt,
                 loss='binary_crossentropy',
                 metrics=['acc'])

    # Early stopping & learning rate decaying
    es_callback = EarlyStopping(monitor='val_acc',patience=15)
    ls_callback = LearningRateScheduler(scheduler)

    # Training
    hist = model.fit(X_train, y_train, epochs=100,
                        validation_data=(X_val,y_val),
                        callbacks=[es_callback,ls_callback])
    # Valiation performance
    print(model.evaluate(X_val, y_val))
```

```
Epoch 1/100
687/687 [==============================] - 6s 6ms/step - loss: 0.5033 - acc:
0.8279 - val_loss: 0.4387 - val_acc: 0.8254 - lr: 0.0100
Epoch 2/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4439 - acc:
0.8296 - val_loss: 0.4138 - val_acc: 0.8402 - lr: 0.0100
```

```
Epoch 3/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4319 - acc:
0.8299 - val_loss: 0.4219 - val_acc: 0.8194 - lr: 0.0100
Epoch 4/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4299 - acc:
0.8304 - val_loss: 0.4034 - val_acc: 0.8395 - lr: 0.0100
Epoch 5/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4304 - acc:
0.8300 - val_loss: 0.4015 - val_acc: 0.8369 - lr: 0.0100
Epoch 6/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4272 - acc:
0.8276 - val_loss: 0.4371 - val_acc: 0.8351 - lr: 0.0100
Epoch 7/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4264 - acc:
0.8283 - val_loss: 0.3949 - val_acc: 0.8421 - lr: 0.0100
Epoch 8/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4243 - acc:
0.8311 - val_loss: 0.4040 - val_acc: 0.8296 - lr: 0.0100
Epoch 9/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4217 - acc:
0.8285 - val_loss: 0.3962 - val_acc: 0.8441 - lr: 0.0100
Epoch 10/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4219 - acc:
0.8310 - val_loss: 0.4159 - val_acc: 0.8318 - lr: 0.0100
Epoch 11/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4233 - acc:
0.8255 - val_loss: 0.4062 - val_acc: 0.8415 - lr: 0.0100
Epoch 12/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4209 - acc:
0.8287 - val_loss: 0.4081 - val_acc: 0.8235 - lr: 0.0100
Epoch 13/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4210 - acc:
0.8297 - val_loss: 0.4230 - val_acc: 0.8223 - lr: 0.0100
Epoch 14/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4205 - acc:
0.8299 - val_loss: 0.3989 - val_acc: 0.8339 - lr: 0.0100
Epoch 15/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4176 - acc:
0.8300 - val_loss: 0.4086 - val_acc: 0.8354 - lr: 0.0100
Epoch 16/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4227 - acc:
0.8270 - val_loss: 0.4172 - val_acc: 0.8234 - lr: 0.0100
Epoch 17/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4181 - acc:
0.8289 - val_loss: 0.3941 - val_acc: 0.8361 - lr: 0.0100
Epoch 18/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4202 - acc:
0.8297 - val_loss: 0.4111 - val_acc: 0.8411 - lr: 0.0100
```

```
Epoch 19/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4186 - acc:
0.8323 - val_loss: 0.3897 - val_acc: 0.8417 - lr: 0.0100
Epoch 20/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4197 - acc:
0.8293 - val_loss: 0.3929 - val_acc: 0.8436 - lr: 0.0100
Epoch 21/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3756 - acc:
0.8405 - val_loss: 0.3378 - val_acc: 0.8503 - lr: 1.0000e-03
Epoch 22/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3556 - acc:
0.8428 - val_loss: 0.3342 - val_acc: 0.8501 - lr: 1.0000e-03
Epoch 23/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3560 - acc:
0.8435 - val_loss: 0.3342 - val_acc: 0.8441 - lr: 1.0000e-03
Epoch 24/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3513 - acc:
0.8470 - val_loss: 0.3337 - val_acc: 0.8497 - lr: 1.0000e-03
Epoch 25/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3500 - acc:
0.8456 - val_loss: 0.3328 - val_acc: 0.8483 - lr: 1.0000e-03
Epoch 26/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3492 - acc:
0.8451 - val_loss: 0.3348 - val_acc: 0.8459 - lr: 1.0000e-03
Epoch 27/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3490 - acc:
0.8455 - val_loss: 0.3313 - val_acc: 0.8489 - lr: 1.0000e-03
Epoch 28/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3485 - acc:
0.8435 - val_loss: 0.3295 - val_acc: 0.8546 - lr: 1.0000e-03
Epoch 29/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3480 - acc:
0.8439 - val_loss: 0.3361 - val_acc: 0.8451 - lr: 1.0000e-03
Epoch 30/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3486 - acc:
0.8460 - val_loss: 0.3356 - val_acc: 0.8471 - lr: 1.0000e-03
Epoch 31/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3522 - acc:
0.8417 - val_loss: 0.3334 - val_acc: 0.8475 - lr: 1.0000e-03
Epoch 32/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3504 - acc:
0.8453 - val_loss: 0.3347 - val_acc: 0.8445 - lr: 1.0000e-03
Epoch 33/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3495 - acc:
0.8425 - val_loss: 0.3299 - val_acc: 0.8512 - lr: 1.0000e-03
Epoch 34/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3508 - acc:
0.8448 - val_loss: 0.3312 - val_acc: 0.8488 - lr: 1.0000e-03
```

```
Epoch 35/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3505 - acc:
0.8437 - val_loss: 0.3327 - val_acc: 0.8490 - lr: 1.0000e-03
Epoch 36/100
687/687 [==============================] - 5s 8ms/step - loss: 0.3488 - acc:
0.8473 - val_loss: 0.3308 - val_acc: 0.8504 - lr: 1.0000e-03
Epoch 37/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3497 - acc:
0.8421 - val_loss: 0.3306 - val_acc: 0.8488 - lr: 1.0000e-03
Epoch 38/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3477 - acc:
0.8452 - val_loss: 0.3306 - val_acc: 0.8496 - lr: 1.0000e-03
Epoch 39/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3503 - acc:
0.8434 - val_loss: 0.3292 - val_acc: 0.8509 - lr: 1.0000e-03
Epoch 40/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3509 - acc:
0.8448 - val_loss: 0.3300 - val_acc: 0.8482 - lr: 1.0000e-03
Epoch 41/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3433 - acc:
0.8477 - val_loss: 0.3273 - val_acc: 0.8501 - lr: 1.0000e-04
Epoch 42/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3396 - acc:
0.8496 - val_loss: 0.3261 - val_acc: 0.8514 - lr: 1.0000e-04
Epoch 43/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3386 - acc:
0.8507 - val_loss: 0.3239 - val_acc: 0.8531 - lr: 1.0000e-04
229/229 [==============================] - 1s 2ms/step - loss: 0.3239 - acc:
0.8531
[0.32385891675949097, 0.8531258702278137]
Epoch 1/100
687/687 [==============================] - 4s 4ms/step - loss: 0.5014 - acc:
0.8285 - val_loss: 0.4381 - val_acc: 0.8332 - lr: 0.0100
Epoch 2/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4441 - acc:
0.8288 - val_loss: 0.4265 - val_acc: 0.8382 - lr: 0.0100
Epoch 3/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4349 - acc:
0.8303 - val_loss: 0.4093 - val_acc: 0.8354 - lr: 0.0100
Epoch 4/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4285 - acc:
0.8311 - val_loss: 0.4174 - val_acc: 0.8361 - lr: 0.0100
Epoch 5/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4320 - acc:
0.8284 - val_loss: 0.4042 - val_acc: 0.8438 - lr: 0.0100
Epoch 6/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4287 - acc:
0.8306 - val_loss: 0.4056 - val_acc: 0.8387 - lr: 0.0100
```

```
Epoch 7/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4206 - acc:
0.8321 - val_loss: 0.4131 - val_acc: 0.8299 - lr: 0.0100
Epoch 8/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4209 - acc:
0.8329 - val_loss: 0.4274 - val_acc: 0.8280 - lr: 0.0100
Epoch 9/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4237 - acc:
0.8300 - val_loss: 0.4046 - val_acc: 0.8438 - lr: 0.0100
Epoch 10/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4187 - acc:
0.8291 - val_loss: 0.4267 - val_acc: 0.8161 - lr: 0.0100
Epoch 11/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4197 - acc:
0.8305 - val_loss: 0.3991 - val_acc: 0.8380 - lr: 0.0100
Epoch 12/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4218 - acc:
0.8276 - val_loss: 0.4005 - val_acc: 0.8381 - lr: 0.0100
Epoch 13/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4176 - acc:
0.8305 - val_loss: 0.3965 - val_acc: 0.8307 - lr: 0.0100
Epoch 14/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4153 - acc:
0.8300 - val_loss: 0.4025 - val_acc: 0.8310 - lr: 0.0100
Epoch 15/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4188 - acc:
0.8299 - val_loss: 0.4045 - val_acc: 0.8279 - lr: 0.0100
Epoch 16/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4152 - acc:
0.8343 - val_loss: 0.3958 - val_acc: 0.8404 - lr: 0.0100
Epoch 17/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4190 - acc:
0.8290 - val_loss: 0.3941 - val_acc: 0.8410 - lr: 0.0100
Epoch 18/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4156 - acc:
0.8281 - val_loss: 0.4001 - val_acc: 0.8430 - lr: 0.0100
Epoch 19/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4155 - acc:
0.8301 - val_loss: 0.3924 - val_acc: 0.8470 - lr: 0.0100
Epoch 20/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4189 - acc:
0.8321 - val_loss: 0.4014 - val_acc: 0.8249 - lr: 0.0100
Epoch 21/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3728 - acc:
0.8421 - val_loss: 0.3439 - val_acc: 0.8479 - lr: 1.0000e-03
Epoch 22/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3549 - acc:
0.8443 - val_loss: 0.3381 - val_acc: 0.8494 - lr: 1.0000e-03
```

```
Epoch 23/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3511 - acc:
0.8468 - val_loss: 0.3365 - val_acc: 0.8508 - lr: 1.0000e-03
Epoch 24/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3523 - acc:
0.8437 - val_loss: 0.3377 - val_acc: 0.8488 - lr: 1.0000e-03
Epoch 25/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3493 - acc:
0.8455 - val_loss: 0.3338 - val_acc: 0.8504 - lr: 1.0000e-03
Epoch 26/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3518 - acc:
0.8419 - val_loss: 0.3395 - val_acc: 0.8479 - lr: 1.0000e-03
Epoch 27/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3483 - acc:
0.8443 - val_loss: 0.3383 - val_acc: 0.8459 - lr: 1.0000e-03
Epoch 28/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3479 - acc:
0.8452 - val_loss: 0.3385 - val_acc: 0.8505 - lr: 1.0000e-03
Epoch 29/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3482 - acc:
0.8475 - val_loss: 0.3362 - val_acc: 0.8493 - lr: 1.0000e-03
Epoch 30/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3481 - acc:
0.8462 - val_loss: 0.3336 - val_acc: 0.8509 - lr: 1.0000e-03
Epoch 31/100
687/687 [==============================] - 4s 7ms/step - loss: 0.3464 - acc:
0.8436 - val_loss: 0.3371 - val_acc: 0.8467 - lr: 1.0000e-03
Epoch 32/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3468 - acc:
0.8452 - val_loss: 0.3344 - val_acc: 0.8496 - lr: 1.0000e-03
Epoch 33/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3479 - acc:
0.8483 - val_loss: 0.3351 - val_acc: 0.8511 - lr: 1.0000e-03
Epoch 34/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3503 - acc:
0.8433 - val_loss: 0.3373 - val_acc: 0.8507 - lr: 1.0000e-03
Epoch 35/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3484 - acc:
0.8435 - val_loss: 0.3350 - val_acc: 0.8489 - lr: 1.0000e-03
Epoch 36/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3442 - acc:
0.8457 - val_loss: 0.3379 - val_acc: 0.8500 - lr: 1.0000e-03
Epoch 37/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3489 - acc:
0.8448 - val_loss: 0.3391 - val_acc: 0.8492 - lr: 1.0000e-03
Epoch 38/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3483 - acc:
0.8451 - val_loss: 0.3352 - val_acc: 0.8478 - lr: 1.0000e-03
```

```
Epoch 39/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3456 - acc:
0.8466 - val_loss: 0.3363 - val_acc: 0.8520 - lr: 1.0000e-03
Epoch 40/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3463 - acc:
0.8453 - val_loss: 0.3351 - val_acc: 0.8492 - lr: 1.0000e-03
Epoch 41/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3436 - acc:
0.8481 - val_loss: 0.3326 - val_acc: 0.8507 - lr: 1.0000e-04
Epoch 42/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3385 - acc:
0.8487 - val_loss: 0.3308 - val_acc: 0.8515 - lr: 1.0000e-04
Epoch 43/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3385 - acc:
0.8496 - val_loss: 0.3300 - val_acc: 0.8516 - lr: 1.0000e-04
Epoch 44/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3369 - acc:
0.8503 - val_loss: 0.3298 - val_acc: 0.8512 - lr: 1.0000e-04
Epoch 45/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3360 - acc:
0.8503 - val_loss: 0.3265 - val_acc: 0.8552 - lr: 1.0000e-04
Epoch 46/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3336 - acc:
0.8504 - val_loss: 0.3279 - val_acc: 0.8507 - lr: 1.0000e-04
Epoch 47/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3345 - acc:
0.8511 - val_loss: 0.3267 - val_acc: 0.8511 - lr: 1.0000e-04
Epoch 48/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3340 - acc:
0.8518 - val_loss: 0.3261 - val_acc: 0.8537 - lr: 1.0000e-04
Epoch 49/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3327 - acc:
0.8533 - val_loss: 0.3257 - val_acc: 0.8518 - lr: 1.0000e-04
Epoch 50/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3307 - acc:
0.8509 - val_loss: 0.3249 - val_acc: 0.8523 - lr: 1.0000e-04
Epoch 51/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3318 - acc:
0.8530 - val_loss: 0.3244 - val_acc: 0.8549 - lr: 1.0000e-04
Epoch 52/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3323 - acc:
0.8521 - val_loss: 0.3243 - val_acc: 0.8548 - lr: 1.0000e-04
Epoch 53/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3343 - acc:
0.8537 - val_loss: 0.3245 - val_acc: 0.8520 - lr: 1.0000e-04
Epoch 54/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3286 - acc:
0.8513 - val_loss: 0.3223 - val_acc: 0.8557 - lr: 1.0000e-04
```

```
Epoch 55/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3280 - acc:
0.8519 - val_loss: 0.3244 - val_acc: 0.8539 - lr: 1.0000e-04
Epoch 56/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3331 - acc:
0.8504 - val_loss: 0.3241 - val_acc: 0.8544 - lr: 1.0000e-04
Epoch 57/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3276 - acc:
0.8533 - val_loss: 0.3237 - val_acc: 0.8545 - lr: 1.0000e-04
Epoch 58/100
687/687 [==============================] - 3s 4ms/step - loss: 0.3313 - acc:
0.8515 - val_loss: 0.3224 - val_acc: 0.8535 - lr: 1.0000e-04
Epoch 59/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3290 - acc:
0.8506 - val_loss: 0.3232 - val_acc: 0.8546 - lr: 1.0000e-04
Epoch 60/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3301 - acc:
0.8514 - val_loss: 0.3215 - val_acc: 0.8552 - lr: 1.0000e-04
Epoch 61/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3304 - acc:
0.8513 - val_loss: 0.3236 - val_acc: 0.8531 - lr: 1.0000e-05
Epoch 62/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3253 - acc:
0.8539 - val_loss: 0.3224 - val_acc: 0.8552 - lr: 1.0000e-05
Epoch 63/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3268 - acc:
0.8536 - val_loss: 0.3234 - val_acc: 0.8531 - lr: 1.0000e-05
Epoch 64/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3285 - acc:
0.8509 - val_loss: 0.3222 - val_acc: 0.8539 - lr: 1.0000e-05
Epoch 65/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3271 - acc:
0.8544 - val_loss: 0.3226 - val_acc: 0.8531 - lr: 1.0000e-05
Epoch 66/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3254 - acc:
0.8536 - val_loss: 0.3214 - val_acc: 0.8545 - lr: 1.0000e-05
Epoch 67/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3306 - acc:
0.8544 - val_loss: 0.3213 - val_acc: 0.8548 - lr: 1.0000e-05
Epoch 68/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3279 - acc:
0.8546 - val_loss: 0.3211 - val_acc: 0.8544 - lr: 1.0000e-05
Epoch 69/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3283 - acc:
0.8545 - val_loss: 0.3216 - val_acc: 0.8546 - lr: 1.0000e-05
229/229 [==============================] - 1s 2ms/step - loss: 0.3216 - acc:
0.8546
[0.3215961158275604, 0.8546273708343506]
```

```
Epoch 1/100
687/687 [==============================] - 4s 5ms/step - loss: 0.5059 - acc:
0.8265 - val_loss: 0.4301 - val_acc: 0.8370 - lr: 0.0100
Epoch 2/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4449 - acc:
0.8276 - val_loss: 0.4575 - val_acc: 0.8201 - lr: 0.0100
Epoch 3/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4355 - acc:
0.8273 - val_loss: 0.4123 - val_acc: 0.8403 - lr: 0.0100
Epoch 4/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4311 - acc:
0.8286 - val_loss: 0.4170 - val_acc: 0.8320 - lr: 0.0100
Epoch 5/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4255 - acc:
0.8317 - val_loss: 0.4163 - val_acc: 0.8421 - lr: 0.0100
Epoch 6/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4247 - acc:
0.8295 - val_loss: 0.4178 - val_acc: 0.8266 - lr: 0.0100
Epoch 7/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4260 - acc:
0.8270 - val_loss: 0.4092 - val_acc: 0.8325 - lr: 0.0100
Epoch 8/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4253 - acc:
0.8276 - val_loss: 0.4127 - val_acc: 0.8406 - lr: 0.0100
Epoch 9/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4206 - acc:
0.8332 - val_loss: 0.4021 - val_acc: 0.8404 - lr: 0.0100
Epoch 10/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4210 - acc:
0.8304 - val_loss: 0.4149 - val_acc: 0.8351 - lr: 0.0100
Epoch 11/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4192 - acc:
0.8313 - val_loss: 0.3971 - val_acc: 0.8447 - lr: 0.0100
Epoch 12/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4194 - acc:
0.8296 - val_loss: 0.4146 - val_acc: 0.8318 - lr: 0.0100
Epoch 13/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4218 - acc:
0.8296 - val_loss: 0.4147 - val_acc: 0.8152 - lr: 0.0100
Epoch 14/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4167 - acc:
0.8340 - val_loss: 0.4008 - val_acc: 0.8399 - lr: 0.0100
Epoch 15/100
687/687 [==============================] - 2s 3ms/step - loss: 0.4201 - acc:
0.8281 - val_loss: 0.4053 - val_acc: 0.8402 - lr: 0.0100
Epoch 16/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4204 - acc:
0.8275 - val_loss: 0.4017 - val_acc: 0.8382 - lr: 0.0100
```

```
Epoch 17/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4184 - acc:
0.8317 - val_loss: 0.3938 - val_acc: 0.8378 - lr: 0.0100
Epoch 18/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4166 - acc:
0.8303 - val_loss: 0.3941 - val_acc: 0.8458 - lr: 0.0100
Epoch 19/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4130 - acc:
0.8327 - val_loss: 0.4013 - val_acc: 0.8382 - lr: 0.0100
Epoch 20/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4148 - acc:
0.8313 - val_loss: 0.4098 - val_acc: 0.8332 - lr: 0.0100
Epoch 21/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3722 - acc:
0.8412 - val_loss: 0.3523 - val_acc: 0.8411 - lr: 1.0000e-03
Epoch 22/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3533 - acc:
0.8458 - val_loss: 0.3407 - val_acc: 0.8474 - lr: 1.0000e-03
Epoch 23/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3475 - acc:
0.8471 - val_loss: 0.3379 - val_acc: 0.8488 - lr: 1.0000e-03
Epoch 24/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3487 - acc:
0.8433 - val_loss: 0.3416 - val_acc: 0.8494 - lr: 1.0000e-03
Epoch 25/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3484 - acc:
0.8444 - val_loss: 0.3364 - val_acc: 0.8497 - lr: 1.0000e-03
Epoch 26/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3454 - acc:
0.8463 - val_loss: 0.3376 - val_acc: 0.8468 - lr: 1.0000e-03
Epoch 27/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3466 - acc:
0.8448 - val_loss: 0.3409 - val_acc: 0.8432 - lr: 1.0000e-03
Epoch 28/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3469 - acc:
0.8473 - val_loss: 0.3394 - val_acc: 0.8478 - lr: 1.0000e-03
Epoch 29/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3482 - acc:
0.8450 - val_loss: 0.3397 - val_acc: 0.8481 - lr: 1.0000e-03
Epoch 30/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3456 - acc:
0.8469 - val_loss: 0.3401 - val_acc: 0.8455 - lr: 1.0000e-03
Epoch 31/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3481 - acc:
0.8436 - val_loss: 0.3376 - val_acc: 0.8468 - lr: 1.0000e-03
Epoch 32/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3472 - acc:
0.8451 - val_loss: 0.3400 - val_acc: 0.8447 - lr: 1.0000e-03
```

```
Epoch 33/100
687/687 [==============================] - 4s 7ms/step - loss: 0.3461 - acc:
0.8467 - val_loss: 0.3375 - val_acc: 0.8473 - lr: 1.0000e-03
Epoch 34/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3476 - acc:
0.8435 - val_loss: 0.3423 - val_acc: 0.8428 - lr: 1.0000e-03
Epoch 35/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3510 - acc:
0.8453 - val_loss: 0.3398 - val_acc: 0.8437 - lr: 1.0000e-03
Epoch 36/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3465 - acc:
0.8475 - val_loss: 0.3379 - val_acc: 0.8437 - lr: 1.0000e-03
Epoch 37/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3472 - acc:
0.8471 - val_loss: 0.3373 - val_acc: 0.8448 - lr: 1.0000e-03
Epoch 38/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3455 - acc:
0.8463 - val_loss: 0.3398 - val_acc: 0.8419 - lr: 1.0000e-03
Epoch 39/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3468 - acc:
0.8439 - val_loss: 0.3390 - val_acc: 0.8441 - lr: 1.0000e-03
Epoch 40/100
687/687 [==============================] - 5s 7ms/step - loss: 0.3452 - acc:
0.8452 - val_loss: 0.3392 - val_acc: 0.8463 - lr: 1.0000e-03
229/229 [==============================] - 1s 4ms/step - loss: 0.3392 - acc:
0.8463
[0.3392113745212555, 0.8463008403778076]
Epoch 1/100
687/687 [==============================] - 5s 6ms/step - loss: 0.5088 - acc:
0.8263 - val_loss: 0.4201 - val_acc: 0.8479 - lr: 0.0100
Epoch 2/100
687/687 [==============================] - 5s 7ms/step - loss: 0.4478 - acc:
0.8303 - val_loss: 0.4263 - val_acc: 0.8428 - lr: 0.0100
Epoch 3/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4396 - acc:
0.8291 - val_loss: 0.4092 - val_acc: 0.8429 - lr: 0.0100
Epoch 4/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4360 - acc:
0.8278 - val_loss: 0.4082 - val_acc: 0.8471 - lr: 0.0100
Epoch 5/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4324 - acc:
0.8270 - val_loss: 0.4072 - val_acc: 0.8501 - lr: 0.0100
Epoch 6/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4275 - acc:
0.8295 - val_loss: 0.3946 - val_acc: 0.8451 - lr: 0.0100
Epoch 7/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4271 - acc:
0.8271 - val_loss: 0.3984 - val_acc: 0.8380 - lr: 0.0100
```

```
Epoch 8/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4237 - acc:
0.8295 - val_loss: 0.4113 - val_acc: 0.8280 - lr: 0.0100
Epoch 9/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4281 - acc:
0.8256 - val_loss: 0.4043 - val_acc: 0.8426 - lr: 0.0100
Epoch 10/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4279 - acc:
0.8286 - val_loss: 0.4044 - val_acc: 0.8445 - lr: 0.0100
Epoch 11/100
687/687 [==============================] - 4s 5ms/step - loss: 0.4243 - acc:
0.8278 - val_loss: 0.3929 - val_acc: 0.8509 - lr: 0.0100
Epoch 12/100
687/687 [==============================] - 3s 4ms/step - loss: 0.4246 - acc:
0.8268 - val_loss: 0.3886 - val_acc: 0.8447 - lr: 0.0100
Epoch 13/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4224 - acc:
0.8305 - val_loss: 0.3978 - val_acc: 0.8448 - lr: 0.0100
Epoch 14/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4201 - acc:
0.8292 - val_loss: 0.3997 - val_acc: 0.8426 - lr: 0.0100
Epoch 15/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4217 - acc:
0.8287 - val_loss: 0.3887 - val_acc: 0.8447 - lr: 0.0100
Epoch 16/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4202 - acc:
0.8296 - val_loss: 0.3858 - val_acc: 0.8496 - lr: 0.0100
Epoch 17/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4216 - acc:
0.8298 - val_loss: 0.3848 - val_acc: 0.8514 - lr: 0.0100
Epoch 18/100
687/687 [==============================] - 4s 6ms/step - loss: 0.4218 - acc:
0.8296 - val_loss: 0.3946 - val_acc: 0.8385 - lr: 0.0100
Epoch 19/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4187 - acc:
0.8301 - val_loss: 0.3998 - val_acc: 0.8437 - lr: 0.0100
Epoch 20/100
687/687 [==============================] - 3s 5ms/step - loss: 0.4166 - acc:
0.8274 - val_loss: 0.3911 - val_acc: 0.8407 - lr: 0.0100
Epoch 21/100
687/687 [==============================] - 3s 4ms/step - loss: 0.3748 - acc:
0.8401 - val_loss: 0.3424 - val_acc: 0.8511 - lr: 1.0000e-03
Epoch 22/100
687/687 [==============================] - 3s 4ms/step - loss: 0.3567 - acc:
0.8419 - val_loss: 0.3370 - val_acc: 0.8519 - lr: 1.0000e-03
Epoch 23/100
687/687 [==============================] - 3s 4ms/step - loss: 0.3533 - acc:
0.8430 - val_loss: 0.3304 - val_acc: 0.8546 - lr: 1.0000e-03
```

```
Epoch 24/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3519 - acc:
0.8445 - val_loss: 0.3309 - val_acc: 0.8533 - lr: 1.0000e-03
Epoch 25/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3498 - acc:
0.8439 - val_loss: 0.3313 - val_acc: 0.8526 - lr: 1.0000e-03
Epoch 26/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3502 - acc:
0.8438 - val_loss: 0.3299 - val_acc: 0.8524 - lr: 1.0000e-03
Epoch 27/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3510 - acc:
0.8417 - val_loss: 0.3311 - val_acc: 0.8531 - lr: 1.0000e-03
Epoch 28/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3515 - acc:
0.8451 - val_loss: 0.3328 - val_acc: 0.8494 - lr: 1.0000e-03
Epoch 29/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3484 - acc:
0.8446 - val_loss: 0.3302 - val_acc: 0.8530 - lr: 1.0000e-03
Epoch 30/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3483 - acc:
0.8456 - val_loss: 0.3287 - val_acc: 0.8538 - lr: 1.0000e-03
Epoch 31/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3502 - acc:
0.8430 - val_loss: 0.3309 - val_acc: 0.8535 - lr: 1.0000e-03
Epoch 32/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3499 - acc:
0.8456 - val_loss: 0.3332 - val_acc: 0.8504 - lr: 1.0000e-03
Epoch 33/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3492 - acc:
0.8455 - val_loss: 0.3301 - val_acc: 0.8526 - lr: 1.0000e-03
Epoch 34/100
687/687 [==============================] - 3s 5ms/step - loss: 0.3494 - acc:
0.8431 - val_loss: 0.3298 - val_acc: 0.8505 - lr: 1.0000e-03
Epoch 35/100
687/687 [==============================] - 4s 6ms/step - loss: 0.3516 - acc:
0.8475 - val_loss: 0.3266 - val_acc: 0.8535 - lr: 1.0000e-03
Epoch 36/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3498 - acc:
0.8453 - val_loss: 0.3292 - val_acc: 0.8519 - lr: 1.0000e-03
Epoch 37/100
687/687 [==============================] - 3s 4ms/step - loss: 0.3466 - acc:
0.8454 - val_loss: 0.3302 - val_acc: 0.8537 - lr: 1.0000e-03
Epoch 38/100
687/687 [==============================] - 4s 5ms/step - loss: 0.3505 - acc:
0.8429 - val_loss: 0.3307 - val_acc: 0.8546 - lr: 1.0000e-03
229/229 [==============================] - 0s 2ms/step - loss: 0.3307 - acc:
0.8546
[0.3306540846824646, 0.8546273708343506]
```

[15]:

```
model.save("2layerDNN")
```

```
INFO:tensorflow:Assets written to: 2layerDNN\assets
```

[16]:

```
from tensorflow.keras.models import load_model
loaded_model = load_model('2layerDNN')
```

[17]:

```
loaded_model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 128)               11776

 batch_normalization_3 (Batc  (None, 128)              512
 hNormalization)

 re_lu_3 (ReLU)              (None, 128)               0

 dropout_3 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 1)                 129

=================================================================
Total params: 12,417
Trainable params: 12,161
Non-trainable params: 256
_________________________________________________________________
```

[ ]:

```

```

[ ]:

```

```