

1. ارزیابی یک مدل یادگیری عمیق با استفاده از دیتاست ارزیابی اتوماتیک.
2. ارزیابی یک مدل یادگیری عمیق با استفاده از دیتاست ارزیابی دستی.
3. k-fold cross validation. ارزیابی یک مدل یادگیری عمیق با.

In [1]: 1 *# بخش بندی دیتا*

روش اتوماتیک

In [1]: 1 **from** keras.models **import** Sequential
2 **from** keras.layers **import** Dense

Using TensorFlow backend.

In [2]: 1 *# وارد کردن کتابخانه*
2 **import** pandas **as** pd
3 **import** numpy **as** np
4
5 *# وارد کردن دیتاست دیابت*
6 dataset = pd.read_excel("C:\\Users\\ShahinN\\Desktop\\pima-indians-diabetes.
7 dataset.head()

Out[2]:

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [3]: 1 *# بخش بندی دیتاست*
2 X = dataset.values[:,0:8]
3 Y = dataset.values[:,8]

In [4]: 1 *# تعریف مدل*
2 model = Sequential()
3 model.add(Dense(12, input_dim=8, kernel_initializer= 'uniform' , activation=
4 model.add(Dense(8, kernel_initializer= 'uniform' , activation= 'relu'))
5 model.add(Dense(1, kernel_initializer= 'uniform' , activation= 'sigmoid'))

In [5]:

```
1 # Compile model
2 model.compile(loss= 'binary_crossentropy' , optimizer= 'adam' , metrics=[ 'a
```

WARNING:tensorflow:From C:\Users\ShahinN\Anaconda3\lib\site-packages\tensorflow\python\ops\nn_impl.py:180: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

In [6]:

```
1 # Fit the model
2 model.fit(X, Y, validation_split=0.33, epochs=150, batch_size=10)
```

WARNING:tensorflow:From C:\Users\ShahinN\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Train on 514 samples, validate on 254 samples

Epoch 1/150

514/514 [=====] - 4s 7ms/step - loss: 0.6843 - accuracy: 0.6148 - val_loss: 0.6663 - val_accuracy: 0.6732

Epoch 2/150

514/514 [=====] - 0s 571us/step - loss: 0.6710 - accuracy: 0.6401 - val_loss: 0.6592 - val_accuracy: 0.6732

Epoch 3/150

514/514 [=====] - 0s 523us/step - loss: 0.6616 - accuracy: 0.6381 - val_loss: 0.6490 - val_accuracy: 0.6811

Epoch 4/150

514/514 [=====] - 0s 452us/step - loss: 0.6591 - accuracy: 0.6323 - val_loss: 0.6451 - val_accuracy: 0.6811

Epoch 5/150

514/514 [=====] - 0s 412us/step - loss: 0.6536 - accuracy: 0.6401 - val_loss: 0.6424 - val_accuracy: 0.6654

روش دستی

In [7]:

```
1 # بخش بندی دیتا : 66% برای آموزش و 33% برای تست
2
3 from sklearn.model_selection import train_test_split
4 X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.33, ra
5
```

In [8]:

```
1 # تعریف مدل
2 model = Sequential()
3 model.add(Dense(12, input_dim=8, kernel_initializer= 'uniform' , activation=
4 model.add(Dense(8, kernel_initializer= 'uniform' , activation= 'relu' ))
5 model.add(Dense(1, kernel_initializer= 'uniform' , activation= 'sigmoid' ))
```

```
In [9]: 1 # Compile model
        2 model.compile(loss= 'binary_crossentropy' , optimizer= 'adam' , metrics=[ 'a
```

```
In [10]: 1 # Fit the model
         2 model.fit(X_train, y_train, validation_data=(X_test,y_test), epochs=150, bat
```

```
Train on 514 samples, validate on 254 samples
Epoch 1/150
514/514 [=====] - 1s 3ms/step - loss: 0.6897 - accur
acy: 0.6304 - val_loss: 0.6776 - val_accuracy: 0.6693
Epoch 2/150
514/514 [=====] - 0s 408us/step - loss: 0.6732 - acc
uracy: 0.6420 - val_loss: 0.6536 - val_accuracy: 0.6693
Epoch 3/150
514/514 [=====] - 0s 364us/step - loss: 0.6648 - acc
uracy: 0.6420 - val_loss: 0.6440 - val_accuracy: 0.6693
Epoch 4/150
514/514 [=====] - ETA: 0s - loss: 0.6628 - accuracy:
0.62 - 0s 369us/step - loss: 0.6575 - accuracy: 0.6479 - val_loss: 0.6369 - v
al_accuracy: 0.6693
Epoch 5/150
514/514 [=====] - 0s 432us/step - loss: 0.6498 - acc
uracy: 0.6479 - val_loss: 0.6333 - val_accuracy: 0.6575
Epoch 6/150
514/514 [=====] - 0s 424us/step - loss: 0.6408 - acc
```

3. ارزیابی یک مدل یادگیری عمیق با k-fold cross validation.

```
In [12]: 1 from sklearn.model_selection import StratifiedKFold
```

In [13]:

```

1  # 10-fold cross validation test تعریف
2  kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
3  cvscores = []
4  for train, test in kfold.split(X, Y):
5
6      # تعریف مدل
7      model = Sequential()
8      model.add(Dense(12, input_dim=8, kernel_initializer= 'uniform' , activation=
9      model.add(Dense(8, kernel_initializer= 'uniform' , activation= 'relu' ))
10     model.add(Dense(1, kernel_initializer= 'uniform' , activation= 'sigmoid'
11
12     # Compile model
13     model.compile(loss= 'binary_crossentropy' , optimizer= 'adam' , metrics=
14
15     # Fit the model
16     model.fit(X[train], Y[train], epochs=150, batch_size=10, verbose=0)
17
18     # ارزیابی مدل
19     scores = model.evaluate(X[test], Y[test], verbose=0)
20     print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
21
22     cvscores.append(scores[1] * 100)
23
24     print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))

```

```

accuracy: 83.12%
accuracy: 79.22%
accuracy: 72.73%
accuracy: 76.62%
accuracy: 74.03%
accuracy: 72.73%
accuracy: 67.53%
accuracy: 71.43%
accuracy: 76.32%
accuracy: 84.21%
75.79% (+/- 4.96%)

```

In []:

1