```
D:¥Work¥Gre¥UTD¥Courses¥Spring_II¥Exams¥Tensorflow_developer¥Python_3.9¥tfexam¥tfExamTest6¥hiu¥Scripts¥python.exe D:¥Work¥Gre¥UTD¥Courses¥Spring_II¥Exams¥Tensorflow_developer¥Python_3.9¥tfexam¥tfExamTest6¥pandass.py
     age  sex  cp  trestbps  chol  ...  oldpeak  slope  ca        thal  target
0     63    1   1       145   233  ...      2.3      3   0       fixed       0
1     67    1   4       160   286  ...      1.5      2   3      normal       1
2     67    1   4       120   229  ...      2.6      2   2  reversible       0
3     37    1   3       130   250  ...      3.5      3   0      normal       0
4     41    0   2       130   204  ...      1.4      1   0      normal       0
..   ...  ...  ..       ...   ...  ...      ...    ...  ..         ...     ...
298   52    1   1       118   186  ...      0.0      2   0       fixed       0
299   43    0   4       132   341  ...      3.0      2   0  reversible       1
300   65    1   4       135   254  ...      2.8      2   1  reversible       1
301   48    1   4       130   256  ...      0.0      1   2  reversible       1
302   63    0   4       150   407  ...      4.0      2   3  reversible       1

[303 rows x 14 columns]
age         int64
sex         int64
cp          int64
```

```
19  trestbps      int64
20  chol          int64
21  fbs           int64
22  restecg       int64
23  thalach       int64
24  exang         int64
25  oldpeak     float64
26  slope         int64
27  ca            int64
28  thal         object
29  target        int64
30  dtype: object
31  Column: age
32  [63 67 37 41 56 62 57 53 44 52 48 54 49 64 58 60 50 66
     43 40 69 59 42 55
33   61 65 51 45 39 68 47 35 29 70 46 77 38 34 74 76 71]
34  Column: sex
35  [1 0]
36  Column: cp
37  [1 4 3 2 0]
38  Column: trestbps
39  [145 160 120 130 140 172 150 110 132 117 135 155 125 104
    180 138 108 128
40   100 200 124  94 122 170 105 165 112 102 152 115 118 101
    126 142 174 134
41   148 178 158 192 129 144 123 136 146 106 156 154 114 164
    ]
42  Column: chol
43  [233 286 229 250 204 236 268 354 254 203 192 294
    256 263 199 168 239 275
44   266 211 283 284 224 206 219 340 226 247 167 230 335
    234 177 276 353 243
45   225 269 267 248 197 360 258 308 245 270 208 264 321
```

```
 274 325 235 257 302
 164 231 141 252 255 183 330 222 217 282 288 220 209
227 261 213 174 281
 198 221 205 309 240 289 318 298 265 564 246 322 299
300 293 277 304 214
 207 160 249 394 212 184 315 409 244 305 195 196 273
126 313 259 200 262
 215 228 193 303 271 210 327 149 201 295 306 178 237
218 223 242 319 166
 180 311 278 232 253 342 169 187 157 176 241 131 175
417 290 172 216 188
 185 326 260 182 307 186 341 407]
Column: fbs
[1 0]
Column: restecg
[2 0 1]
Column: thalach
[150 108 129 187 172 178 160 163 147 155 148 153 142 173
 162 174 168 139
 171 144 132 158 114 151 161 179 120 112 137  99 177 141
180 111 143 182
 156 149 145 146 175  97 165 133 126 170 154 202 186 125
 103 130 166 164
 159 184 131 152 124 122  96 109 138 157  88 105 194 195
106 115 167  95
 169 192 117 121 116  71 118 140 181 134 136  90 123 128
188 113 185 190
 127]
Column: exang
[0 1]
Column: oldpeak
[2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 3.1 0.4 1.3 0.  0.5 1.6 1.  1.2
0.2 1.8
```

```
 3.2 2.4 2.  2.5 2.8 3.  6.2 5.6 4.  2.2 2.9 0.1 2.1 1.9 4.2 0.9
1.1 3.8
 0.7 3.4 0.3 4.4]
Column: slope
[3 2 1]
Column: ca
[0 3 2 1]
Column: thal
['fixed' 'normal' 'reversible' '1' '2']
Binary Column: sex
Binary Column: fbs
Binary Column: exang
tf.Tensor(
[[ 0.93383914  0.03480718  0.74578077 -0.26008666  1.
0680453 ]
 [ 1.3782105  -1.7806165   1.5923285   0.7573878   0.
38022864]
 [ 1.3782105  -0.87290466 -0.6651321  -0.3368772   1.
3259765 ]], shape=(3, 5), dtype=float32)
Epoch 1/15
152/152
 [==============================] - 1s
2ms/step - loss: 0.7776 - accuracy: 0.6271
WARNING:tensorflow:Early stopping conditioned on
metric `val_loss` which is not available. Available
metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with
val_loss available, skipping.
Epoch 2/15
152/152
 [==============================] - 0s
2ms/step - loss: 0.7753 - accuracy: 0.6337
Epoch 3/15
```

```
 89   1/152 [..............................] - ETA: 0s - loss: 0.8686 -
      accuracy: 0.5000WARNING:tensorflow:Early stopping
      conditioned on metric `val_loss` which is not available
      . Available metrics are: loss,accuracy
 90   WARNING:tensorflow:Can save best model only with
      val_loss available, skipping.
 91   152/152
       [==============================] - 0s
      2ms/step - loss: 0.7730 - accuracy: 0.6370
 92   WARNING:tensorflow:Early stopping conditioned on
      metric `val_loss` which is not available. Available
      metrics are: loss,accuracy
 93   WARNING:tensorflow:Can save best model only with
      val_loss available, skipping.
 94   Epoch 4/15
 95   152/152
       [==============================] - 0s
      2ms/step - loss: 0.7708 - accuracy: 0.6403
 96   WARNING:tensorflow:Early stopping conditioned on
      metric `val_loss` which is not available. Available
      metrics are: loss,accuracy
 97   WARNING:tensorflow:Can save best model only with
      val_loss available, skipping.
 98   Epoch 5/15
 99   152/152
       [==============================] - 0s
      2ms/step - loss: 0.7686 - accuracy: 0.6436
100   WARNING:tensorflow:Early stopping conditioned on
      metric `val_loss` which is not available. Available
      metrics are: loss,accuracy
101   WARNING:tensorflow:Can save best model only with
      val_loss available, skipping.
102   Epoch 6/15
```

```
152/152 [==============================] - 0s 2ms/step - loss: 0.7665 - accuracy: 0.6469
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
Epoch 7/15
152/152 [==============================] - 0s 2ms/step - loss: 0.7644 - accuracy: 0.6469
Epoch 8/15
  1/152 [..............................] - ETA: 0s - loss: 0.7194 - accuracy: 0.5000WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
152/152 [==============================] - 0s 2ms/step - loss: 0.7623 - accuracy: 0.6535
Epoch 9/15
  1/152 [..............................] - ETA: 0s - loss: 0.5943 - accuracy: 1.0000WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
152/152 [==============================] - 0s 2ms/step - loss: 0.7603 - accuracy: 0.6568
WARNING:tensorflow:Early stopping conditioned on
```

```
116 metric `val_loss` which is not available. Available
    metrics are: loss,accuracy
117 WARNING:tensorflow:Can save best model only with
    val_loss available, skipping.
118 Epoch 10/15
119 152/152
     [==============================] - 0s
    2ms/step - loss: 0.7583 - accuracy: 0.6568
120 Epoch 11/15
121   1/152 [..............................] - ETA: 0s - loss: 0.5401 -
    accuracy: 0.5000WARNING:tensorflow:Early stopping
    conditioned on metric `val_loss` which is not available
    . Available metrics are: loss,accuracy
122 WARNING:tensorflow:Can save best model only with
    val_loss available, skipping.
123 152/152
     [==============================] - 0s
    2ms/step - loss: 0.7563 - accuracy: 0.6601
124 Epoch 12/15
125   1/152 [..............................] - ETA: 0s - loss: 0.7688 -
    accuracy: 0.5000WARNING:tensorflow:Early stopping
    conditioned on metric `val_loss` which is not available
    . Available metrics are: loss,accuracy
126 WARNING:tensorflow:Can save best model only with
    val_loss available, skipping.
127 152/152
     [==============================] - 0s
    2ms/step - loss: 0.7544 - accuracy: 0.6634
128 Epoch 13/15
129 WARNING:tensorflow:Early stopping conditioned on
    metric `val_loss` which is not available. Available
    metrics are: loss,accuracy
130 WARNING:tensorflow:Can save best model only with
```

```
val_loss available, skipping.
152/152
 [==============================] - 0s 2ms/step - loss: 0.7525 - accuracy: 0.6667
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
Epoch 14/15
152/152
 [==============================] - 0s 2ms/step - loss: 0.7507 - accuracy: 0.6667
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
Epoch 15/15
152/152
 [==============================] - 0s 2ms/step - loss: 0.7488 - accuracy: 0.6700
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss,accuracy
WARNING:tensorflow:Can save best model only with val_loss available, skipping.
(<tf.Tensor: shape=(5,), dtype=float64, numpy=array([ 63. , 150. , 145. , 233. ,   2.3])>, <tf.Tensor: shape=(), dtype=int64, numpy=0>)
(<tf.Tensor: shape=(5,), dtype=float64, numpy=array([ 67. , 108. , 160. , 286. ,   1.5])>, <tf.Tensor: shape=(), dtype=int64, numpy=1>)
```

```
(<tf.Tensor: shape=(5,), dtype=float64, numpy=array
([ 67. , 129. , 120. , 229. ,   2.6])>, <tf.Tensor: shape=(),
dtype=int64, numpy=0>)
Epoch 1/15
152/152
 [==============================] - 1s
2ms/step - loss: 0.8789 - accuracy: 0.5347
Epoch 2/15
152/152
 [==============================] - 0s
2ms/step - loss: 0.8750 - accuracy: 0.5314
Epoch 3/15
152/152
 [==============================] - 0s
2ms/step - loss: 0.8711 - accuracy: 0.5413
Epoch 4/15
152/152
 [==============================] - 0s
1ms/step - loss: 0.8674 - accuracy: 0.5446
Epoch 5/15
152/152
 [==============================] - 0s
1ms/step - loss: 0.8636 - accuracy: 0.5446
Epoch 6/15
152/152
 [==============================] - 0s
2ms/step - loss: 0.8599 - accuracy: 0.5479
Epoch 7/15
152/152
 [==============================] - 0s
1ms/step - loss: 0.8562 - accuracy: 0.5512
Epoch 8/15
152/152
```

```
160  [==============================] - 0s
     2ms/step - loss: 0.8526 - accuracy: 0.5545
161  Epoch 9/15
162  152/152
     [==============================] - 0s
     2ms/step - loss: 0.8490 - accuracy: 0.5578
163  Epoch 10/15
164  152/152
     [==============================] - 0s
     1ms/step - loss: 0.8455 - accuracy: 0.5611
165  Epoch 11/15
166  152/152
     [==============================] - 0s
     2ms/step - loss: 0.8421 - accuracy: 0.5578
167  Epoch 12/15
168  152/152
     [==============================] - 0s
     1ms/step - loss: 0.8386 - accuracy: 0.5611
169  Epoch 13/15
170  152/152
     [==============================] - 0s
     1ms/step - loss: 0.8353 - accuracy: 0.5545
171  Epoch 14/15
172  152/152
     [==============================] - 0s
     2ms/step - loss: 0.8319 - accuracy: 0.5611
173  Epoch 15/15
174  152/152
     [==============================] - 0s
     2ms/step - loss: 0.8286 - accuracy: 0.5611
175  ({'age': <tf.Tensor: shape=(), dtype=int64, numpy=63
     >, 'thalach': <tf.Tensor: shape=(), dtype=int64, numpy
     =150>, 'trestbps': <tf.Tensor: shape=(), dtype=int64,
```

```
numpy=145>, 'chol': <tf.Tensor: shape=(), dtype=
int64, numpy=233>, 'oldpeak': <tf.Tensor: shape=(),
dtype=float64, numpy=2.3>}, <tf.Tensor: shape=(),
dtype=int64, numpy=0>)
({'age': <tf.Tensor: shape=(), dtype=int64, numpy=67
>, 'thalach': <tf.Tensor: shape=(), dtype=int64, numpy
=108>, 'trestbps': <tf.Tensor: shape=(), dtype=int64,
numpy=160>, 'chol': <tf.Tensor: shape=(), dtype=
int64, numpy=286>, 'oldpeak': <tf.Tensor: shape=(),
dtype=float64, numpy=1.5>}, <tf.Tensor: shape=(),
dtype=int64, numpy=1>)
({'age': <tf.Tensor: shape=(), dtype=int64, numpy=67
>, 'thalach': <tf.Tensor: shape=(), dtype=int64, numpy
=129>, 'trestbps': <tf.Tensor: shape=(), dtype=int64,
numpy=120>, 'chol': <tf.Tensor: shape=(), dtype=
int64, numpy=229>, 'oldpeak': <tf.Tensor: shape=(),
dtype=float64, numpy=2.6>}, <tf.Tensor: shape=(),
dtype=int64, numpy=0>)
{'age': <KerasTensor: shape=(None, 1) dtype=float32 (
created by layer 'age')>, 'thalach': <KerasTensor:
shape=(None, 1) dtype=float32 (created by layer '
thalach')>, 'trestbps': <KerasTensor: shape=(None, 1)
dtype=float32 (created by layer 'trestbps')>, 'chol': <
KerasTensor: shape=(None, 1) dtype=float32 (created
by layer 'chol')>, 'oldpeak': <KerasTensor: shape=(
None, 1) dtype=float32 (created by layer 'oldpeak')>}
Epoch 1/5
152/152
 [==============================] - 4s
24ms/step - loss: 0.7189 - accuracy: 0.7063
Epoch 2/5
152/152
 [==============================] - 3s
```

```
22ms/step - loss: 0.5866 - accuracy: 0.7525
Epoch 3/5
152/152
 [==============================] - 3s
20ms/step - loss: 0.5044 - accuracy: 0.7525
Epoch 4/5
152/152
 [==============================] - 3s
22ms/step - loss: 0.4613 - accuracy: 0.7558
Epoch 5/5
152/152
 [==============================] - 3s
23ms/step - loss: 0.4446 - accuracy: 0.7690
1/1 [==============================] -
0s 28ms/step
 The prediction of first three rows : [[-0.00848308]
 [ 0.02018291]
 [ 0.02266869]]
{'age': <KerasTensor: shape=(None,) dtype=float32 (
created by layer 'age')>, 'sex': <KerasTensor: shape=(
None,) dtype=int64 (created by layer 'sex')>, 'cp': <
KerasTensor: shape=(None,) dtype=int64 (created by
layer 'cp')>, 'trestbps': <KerasTensor: shape=(None,)
dtype=float32 (created by layer 'trestbps')>, 'chol': <
KerasTensor: shape=(None,) dtype=float32 (created
by layer 'chol')>, 'fbs': <KerasTensor: shape=(None,)
dtype=int64 (created by layer 'fbs')>, 'restecg': <
KerasTensor: shape=(None,) dtype=int64 (created by
layer 'restecg')>, 'thalach': <KerasTensor: shape=(
None,) dtype=float32 (created by layer 'thalach')>, '
exang': <KerasTensor: shape=(None,) dtype=int64 (
created by layer 'exang')>, 'oldpeak': <KerasTensor:
shape=(None,) dtype=float32 (created by layer '
```

```
oldpeak')>, 'slope': <KerasTensor: shape=(None,)
dtype=int64 (created by layer 'slope')>, 'ca': <
KerasTensor: shape=(None,) dtype=int64 (created by
layer 'ca')>, 'thal': <KerasTensor: shape=(None,) dtype
=string (created by layer 'thal')>}
[<KerasTensor: shape=(None, 1) dtype=float32 (
created by layer 'tf.cast_5')>, <KerasTensor: shape=(
None, 1) dtype=float32 (created by layer 'tf.cast_6
')>, <KerasTensor: shape=(None, 1) dtype=float32 (
created by layer 'tf.cast_7')>]
[<KerasTensor: shape=(None, 1) dtype=float32 (
created by layer 'tf.cast_5')>, <KerasTensor: shape=(
None, 1) dtype=float32 (created by layer 'tf.cast_6
')>, <KerasTensor: shape=(None, 1) dtype=float32 (
created by layer 'tf.cast_7')>, <KerasTensor: shape=(
None, 5) dtype=float32 (created by layer '
normalization_2')>]
tf.Tensor(
[[0. 0. 0. 1.]
 [0. 1. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [1. 0. 0. 0.]], shape=(5, 4), dtype=float32)
tf.Tensor(
[[1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]], shape=(3, 5), dtype=float32)
name: cp
vocab: [0, 1, 2, 3, 4]

name: restecg
vocab: [0, 1, 2]

```

```
name: slope
vocab: [1, 2, 3]

name: thal
vocab: ['1', '2', 'fixed', 'normal', 'reversible']

name: ca
vocab: [0, 1, 2, 3]

Final preprocessed head: [<KerasTensor: shape=(
    None, 1) dtype=float32 (created by layer 'tf.cast_5
    ')>, <KerasTensor: shape=(None, 1) dtype=float32 (
    created by layer 'tf.cast_6')>, <KerasTensor: shape=(
    None, 1) dtype=float32 (created by layer 'tf.cast_7
    ')>, <KerasTensor: shape=(None, 5) dtype=float32 (
    created by layer 'normalization_2')>, <KerasTensor:
    shape=(None, 6) dtype=float32 (created by layer '
    integer_lookup_1')>, <KerasTensor: shape=(None, 4)
    dtype=float32 (created by layer 'integer_lookup_2
    ')>, <KerasTensor: shape=(None, 4) dtype=float32 (
    created by layer 'integer_lookup_3')>, <KerasTensor:
    shape=(None, 6) dtype=float32 (created by layer '
    string_lookup_1')>, <KerasTensor: shape=(None, 5)
    dtype=float32 (created by layer 'integer_lookup_4')>]
{'age': <KerasTensor: shape=(None,) dtype=float32 (
    created by layer 'age')>, 'sex': <KerasTensor: shape=(
    None,) dtype=int64 (created by layer 'sex')>, 'cp': <
    KerasTensor: shape=(None,) dtype=int64 (created by
    layer 'cp')>, 'trestbps': <KerasTensor: shape=(None,)
    dtype=float32 (created by layer 'trestbps')>, 'chol': <
    KerasTensor: shape=(None,) dtype=float32 (created
    by layer 'chol')>, 'fbs': <KerasTensor: shape=(None,)
    dtype=int64 (created by layer 'fbs')>, 'restecg': <
```

```
222 KerasTensor: shape=(None,) dtype=int64 (created by
    layer 'restecg')>, 'thalach': <KerasTensor: shape=(
    None,) dtype=float32 (created by layer 'thalach')>, '
    exang': <KerasTensor: shape=(None,) dtype=int64 (
    created by layer 'exang')>, 'oldpeak': <KerasTensor:
    shape=(None,) dtype=float32 (created by layer '
    oldpeak')>, 'slope': <KerasTensor: shape=(None,)
    dtype=int64 (created by layer 'slope')>, 'ca': <
    KerasTensor: shape=(None,) dtype=int64 (created by
    layer 'ca')>, 'thal': <KerasTensor: shape=(None,) dtype
    =string (created by layer 'thal')>}
223 KerasTensor(type_spec=TensorSpec(shape=(None, 1),
    dtype=tf.float32, name=None), name='sequential_2/
    dense_11/BiasAdd:0', description="created by layer '
    sequential_2'")
224
225 Epoch 1: LearningRateScheduler setting learning rate
    to 1e-05.
226 Epoch 1/5
227 109/121 [==========================>...] -
    ETA: 0s - loss: 0.7522 - accuracy: 0.6789
228 Epoch 1: val_loss improved from inf to 0.77778, saving
    model to ./Models/mnist_tfds¥mnist_h5.h5
229 D:¥Work¥Gre¥UTD¥Courses¥Spring_II¥Exams¥
    Tensorflow_developer¥Python_3.9¥tfexam¥
    tfExamTest6¥hiu¥lib¥site-packages¥keras¥src¥engine
    ¥training.py:3000: UserWarning: You are saving your
    model as an HDF5 file via `model.save()`. This file
    format is considered legacy. We recommend using
    instead the native Keras format, e.g. `model.save('
    my_model.keras')`.
230   saving_api.save_model(
231 121/121
```

```
231  [==============================] - 2s
     6ms/step - loss: 0.7530 - accuracy: 0.6860 - val_loss: 0.
     7778 - val_accuracy: 0.5574 - lr: 1.0000e-05
232
233  Epoch 2: LearningRateScheduler setting learning rate
     to 3.1622776601683795e-05.
234  Epoch 2/5
235  114/121
     [============================>..] - ETA:
     0s - loss: 0.7481 - accuracy: 0.6842
236  Epoch 2: val_loss improved from 0.77778 to 0.77261,
     saving model to ./Models/mnist_tfds¥mnist_h5.h5
237  121/121
     [==============================] - 1s
     4ms/step - loss: 0.7499 - accuracy: 0.6901 - val_loss: 0.
     7726 - val_accuracy: 0.5574 - lr: 3.1623e-05
238
239  Epoch 3: LearningRateScheduler setting learning rate
     to 0.0001.
240  Epoch 3/5
241  105/121 [=========================>....] -
     ETA: 0s - loss: 0.7442 - accuracy: 0.6857
242  Epoch 3: val_loss improved from 0.77261 to 0.75861,
     saving model to ./Models/mnist_tfds¥mnist_h5.h5
243  121/121
     [==============================] - 0s
     4ms/step - loss: 0.7412 - accuracy: 0.6942 - val_loss: 0.
     7586 - val_accuracy: 0.5738 - lr: 1.0000e-04
244
245  Epoch 4: LearningRateScheduler setting learning rate
     to 0.00031622776601683794.
246  Epoch 4/5
247  106/121 [=========================>....] -
```

```
247  ETA: 0s - loss: 0.7234 - accuracy: 0.7406
248  Epoch 4: val_loss improved from 0.75861 to 0.72763,
     saving model to ./Models/mnist_tfds¥mnist_h5.h5
249  121/121
      [==============================] - 0s
     3ms/step - loss: 0.7211 - accuracy: 0.7190 - val_loss: 0.
     7276 - val_accuracy: 0.6721 - lr: 3.1623e-04
250
 251 Epoch 5: LearningRateScheduler setting learning rate
     to 0.001.
252  Epoch 5/5
253  111/121 [============================>...] -
     ETA: 0s - loss: 0.6857 - accuracy: 0.7477
254  Epoch 5: val_loss improved from 0.72763 to 0.68504,
     saving model to ./Models/mnist_tfds¥mnist_h5.h5
255  121/121
      [==============================] - 0s
     4ms/step - loss: 0.6829 - accuracy: 0.7438 - val_loss: 0.
     6850 - val_accuracy: 0.6885 - lr: 0.0010
256
257  <keras.src.callbacks.History object at
     0x000002CA2745F790>
258  1/1 [==============================] -
     0s 294ms/step
259  1/1 [==============================] -
     0s 34ms/step
260  1/1 [==============================] -
     0s 32ms/step
 261 Patient 1 is safe
262  The features for the Patient 1 is is: {'age': 243    45
263  Name: age, dtype: int64, 'sex': 243    1
264  Name: sex, dtype: int64, 'cp': 243    1
265  Name: cp, dtype: int64, 'trestbps': 243    110
```

```
Name: trestbps, dtype: int64, 'chol': 243    264
Name: chol, dtype: int64, 'fbs': 243    0
Name: fbs, dtype: int64, 'restecg': 243    0
Name: restecg, dtype: int64, 'thalach': 243    132
Name: thalach, dtype: int64, 'exang': 243    0
Name: exang, dtype: int64, 'oldpeak': 243    1.2
Name: oldpeak, dtype: float64, 'slope': 243    2
Name: slope, dtype: int64, 'ca': 243    0
Name: ca, dtype: int64, 'thal': 243    reversible
Name: thal, dtype: object}
Patient 2 has chances of heart attack
The features for the Patient 2 is is: {'age': 213    59
70    63
Name: age, dtype: int64, 'sex': 213    1
70    1
Name: sex, dtype: int64, 'cp': 213    3
70    4
Name: cp, dtype: int64, 'trestbps': 213    126
70    130
Name: trestbps, dtype: int64, 'chol': 213    218
70    330
Name: chol, dtype: int64, 'fbs': 213    1
70    1
Name: fbs, dtype: int64, 'restecg': 213    0
70    2
Name: restecg, dtype: int64, 'thalach': 213    134
70    132
Name: thalach, dtype: int64, 'exang': 213    0
70    1
Name: exang, dtype: int64, 'oldpeak': 213    2.2
70    1.8
Name: oldpeak, dtype: float64, 'slope': 213    2
70    1
```

```
Name: slope, dtype: int64, 'ca': 213    1
70     3
Name: ca, dtype: int64, 'thal': 213         fixed
70     reversible
Name: thal, dtype: object}
Patient 3 has chances of heart attack
The features for the Patient 3 is is: {'age': 217    66
232    58
207    44
Name: age, dtype: int64, 'sex': 217    1
232    0
207    0
Name: sex, dtype: int64, 'cp': 217    4
232    4
207    3
Name: cp, dtype: int64, 'trestbps': 217    160
232    170
207    118
Name: trestbps, dtype: int64, 'chol': 217    228
232    225
207    242
Name: chol, dtype: int64, 'fbs': 217    0
232    1
207    0
Name: fbs, dtype: int64, 'restecg': 217    2
232    2
207    0
Name: restecg, dtype: int64, 'thalach': 217    138
232    146
207    149
Name: thalach, dtype: int64, 'exang': 217    0
232    1
207    0
```

```
Name: exang, dtype: int64, 'oldpeak': 217    2.3
232    2.8
207    0.3
Name: oldpeak, dtype: float64, 'slope': 217    1
232    2
207    2
Name: slope, dtype: int64, 'ca': 217    0
232    2
207    1
Name: ca, dtype: int64, 'thal': 217     fixed
232     fixed
207    normal
Name: thal, dtype: object}

Process finished with exit code 0

```