```
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
trestbps         int64
chol             int64
fbs              int64
restecg          int64
thalach          int64
exang            int64
oldpeak        float64
```

```
slope         int64
ca            int64
thal         object
target        int64
dtype: object
Column: age
[63 67 37 41 56 62 57 53 44 52 48 54 49 64 58 60 50 66 43 40 69 59 42 55
 61 65 51 45 39 68 47 35 29 70 46 77 38 34 74 76 71]
Column: sex
[1 0]
Column: cp
[1 4 3 2 0]
Column: trestbps
[145 160 120 130 140 172 150 110 132 117 135 155 125 104 180 138 108 128
 100 200 124  94 122 170 105 165 112 102 152 115 118 101 126 142 174 134
 148 178 158 192 129 144 123 136 146 106 156 154 114 164]
Column: chol
[233 286 229 250 204 236 268 354 254 203 192 294 256 263 199 168 239 275
 266 211 283 284 224 206 219 340 226 247 167 230 335 234 177 276 353 243
 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: 0.6538 - accuracy: 0.7261
Epoch 2/15
```

```
  1/152 [..............................] - ETA: 0s - loss: 0.6029 - 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: 0.6518 - accuracy: 0.7261
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 3/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6499 - accuracy: 0.7261
Epoch 4/15
  1/152 [..............................] - ETA: 0s - loss: 0.4892 - 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: 0.6479 - accuracy: 0.7261
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 5/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6460 - accuracy: 0.7261
Epoch 6/15
  1/152 [..............................] - ETA: 0s - loss: 0.4271 - 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: 0.6440 - accuracy: 0.7261
Epoch 7/15
  1/152 [..............................] - ETA: 0s - loss: 0.7872 - 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: 0.6421 - accuracy: 0.7261
Epoch 8/15
  1/152 [..............................] - ETA: 0s - loss: 0.4459 - 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: 0.6401 - accuracy: 0.7261
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 9/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6381 - accuracy: 0.7261
Epoch 10/15
  1/152 [..............................] - ETA: 0s - loss: 0.3419 - 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: 0.6362 - accuracy: 0.7261
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 11/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6342 - accuracy: 0.7261
Epoch 12/15
  1/152 [..............................] - ETA: 0s - loss: 0.5331 - 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: 0.6323 - accuracy: 0.7261
Epoch 13/15
  1/152 [..............................] - ETA: 0s - loss: 0.6775 - 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: 0.6305 - accuracy: 0.7261
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: 0.6286 - accuracy: 0.7261
Epoch 15/15
  1/152 [..............................] - ETA: 0s - loss: 0.3378 - 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: 0.6267 - accuracy: 0.7261
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: 0.6448 - accuracy: 0.7228
Epoch 2/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6433 - accuracy: 0.7228
Epoch 3/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6419 - accuracy: 0.7261
Epoch 4/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6405 - accuracy: 0.7261
Epoch 5/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6391 - accuracy: 0.7261
Epoch 6/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6377 - accuracy: 0.7261
Epoch 7/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6364 - accuracy: 0.7261
```

```
Epoch 8/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6350 - accuracy: 0.7261
Epoch 9/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6337 - accuracy: 0.7261
Epoch 10/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6324 - accuracy: 0.7261
Epoch 11/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6311 - accuracy: 0.7261
Epoch 12/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6298 - accuracy: 0.7261
Epoch 13/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6285 - accuracy: 0.7261
Epoch 14/15
152/152 [==============================] - 0s 1ms/step - loss: 0.6273 - accuracy: 0.7261
Epoch 15/15
152/152 [==============================] - 0s 2ms/step - loss: 0.6260 - accuracy: 0.7261
({'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 [==============================] - 3s 20ms/step - loss: 0.5840 - accuracy: 0.7294
Epoch 2/5
152/152 [==============================] - 3s 20ms/step - loss: 0.5119 - accuracy: 0.7261
Epoch 3/5
152/152 [==============================] - 3s 20ms/step - loss: 0.4862 - accuracy: 0.7261
Epoch 4/5
152/152 [==============================] - 3s 21ms/step - loss: 0.4729 - accuracy: 0.7261
Epoch 5/5
152/152 [==============================] - 3s 22ms/step - loss: 0.4656 - accuracy: 0.7261
1/1 [==============================] - 0s 24ms/step
 The prediction of first three rows : [[-0.13269539]
 [-0.15111043]
 [-0.01944213]]
{'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': <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(type_spec=TensorSpec(shape=(None, 1), dtype=tf.float32, name=None), name='sequential_2/dense_11/BiasAdd:0', description="created by layer 'sequential_2'")

Epoch 1: LearningRateScheduler setting learning rate to 1e-05.
Epoch 1/5
115/121 [===========================>..] - ETA: 0s - loss: 0.6738 - accuracy: 0.7435
Epoch 1: val_loss improved from inf to 0.66843, saving model to ./Models/mnist_tfds\mnist_h5.h5
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')`.
  saving_api.save_model(
121/121 [==============================] - 1s 6ms/step - loss: 0.6783 - accuracy: 0.7438 - val_loss: 0.6684 - val_accuracy: 0.7377 - lr: 1.0000e-05

Epoch 2: LearningRateScheduler setting learning rate to 3.1622776601683795e-05.
Epoch 2/5
 98/121 [=======================>......] - ETA: 0s - loss: 0.6695 - accuracy: 0.7449
Epoch 2: val_loss improved from 0.66843 to 0.66505, saving model to ./Models/mnist_tfds\mnist_h5.h5
121/121 [==============================] - 0s 4ms/step - loss: 0.6755 - accuracy: 0.7438 - val_loss: 0.6650 - val_accuracy: 0.7213 - lr: 3.1623e-05

Epoch 3: LearningRateScheduler setting learning rate
```

```
 to 0.0001.
Epoch 3/5
112/121 [===========================>...] - ETA: 0s - loss: 0.6668 - accuracy: 0.7455
Epoch 3: val_loss improved from 0.66505 to 0.65488, saving model to ./Models/mnist_tfds\mnist_h5.h5
121/121 [==============================] - 0s 4ms/step - loss: 0.6672 - accuracy: 0.7355 - val_loss: 0.6549 - val_accuracy: 0.7213 - lr: 1.0000e-04

Epoch 4: LearningRateScheduler setting learning rate to 0.00031622776601683794.
Epoch 4/5
110/121 [==========================>...] - ETA: 0s - loss: 0.6480 - accuracy: 0.7273
Epoch 4: val_loss improved from 0.65488 to 0.62695, saving model to ./Models/mnist_tfds\mnist_h5.h5
121/121 [==============================] - 0s 4ms/step - loss: 0.6430 - accuracy: 0.7397 - val_loss: 0.6270 - val_accuracy: 0.7213 - lr: 3.1623e-04

Epoch 5: LearningRateScheduler setting learning rate to 0.001.
Epoch 5/5
112/121 [===========================>...] - ETA: 0s - loss: 0.5520 - accuracy: 0.7500
Epoch 5: val_loss improved from 0.62695 to 0.50634, saving model to ./Models/mnist_tfds\mnist_h5.h5
121/121 [==============================] - 0s 4ms/step - loss: 0.5589 - accuracy: 0.7355 - val_loss: 0.5063 - val_accuracy: 0.7377 - lr: 0.0010

<keras.src.callbacks.History object at 0x0000015680F9F400>
1/1 [==============================] - 0s 251ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 47ms/step
Patient 1 is safe
The features for the Patient 1 is is: {'age': 102    55
Name: age, dtype: int64, 'sex': 102    1
Name: sex, dtype: int64, 'cp': 102    4
Name: cp, dtype: int64, 'trestbps': 102    160
Name: trestbps, dtype: int64, 'chol': 102    289
Name: chol, dtype: int64, 'fbs': 102    0
Name: fbs, dtype: int64, 'restecg': 102    2
Name: restecg, dtype: int64, 'thalach': 102    145
Name: thalach, dtype: int64, 'exang': 102    1
Name: exang, dtype: int64, 'oldpeak': 102    0.8
Name: oldpeak, dtype: float64, 'slope': 102    2
Name: slope, dtype: int64, 'ca': 102    1
Name: ca, dtype: int64, 'thal': 102    reversible
Name: thal, dtype: object}
Patient 2 is safe
The features for the Patient 2 is is: {'age': 107    60
87     35
Name: age, dtype: int64, 'sex': 107    1
87     1
Name: sex, dtype: int64, 'cp': 107    4
87     4
Name: cp, dtype: int64, 'trestbps': 107    140
87     120
Name: trestbps, dtype: int64, 'chol': 107    293
87     198
Name: chol, dtype: int64, 'fbs': 107    0
87     0
Name: fbs, dtype: int64, 'restecg': 107    2
87     0
Name: restecg, dtype: int64, 'thalach': 107    170
87     130
Name: thalach, dtype: int64, 'exang': 107    0
87     1
Name: exang, dtype: int64, 'oldpeak': 107    1.2
87     1.6
Name: oldpeak, dtype: float64, 'slope': 107    2
87     2
Name: slope, dtype: int64, 'ca': 107    2
87     0
Name: ca, dtype: int64, 'thal': 107    reversible
87     reversible
Name: thal, dtype: object}
```

```
Patient 3 is safe
The features for the Patient 3 is is: {'age': 52    53
293    57
45     44
Name: age, dtype: int64, 'sex': 52     0
293    1
45     1
Name: sex, dtype: int64, 'cp': 52     4
293    3
45     4
Name: cp, dtype: int64, 'trestbps': 52     130
293    128
45     110
Name: trestbps, dtype: int64, 'chol': 52     264
293    229
45     197
Name: chol, dtype: int64, 'fbs': 52     0
293    0
45     0
Name: fbs, dtype: int64, 'restecg': 52     2
293    2
45     2
Name: restecg, dtype: int64, 'thalach': 52     143
293    150
45     177
Name: thalach, dtype: int64, 'exang': 52     0
293    0
45     0
Name: exang, dtype: int64, 'oldpeak': 52     0.4
293    0.4
45     0.0
Name: oldpeak, dtype: float64, 'slope': 52     2
293    2
45     1
Name: slope, dtype: int64, 'ca': 52     0
293    1
45     1
Name: ca, dtype: int64, 'thal': 52         normal
293    reversible
45         normal
Name: thal, dtype: object}

Process finished with exit code 0

```