In [1]:
```python
import numpy as np
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
import keras
import tensorflow as tf
from keras.utils import to_categorical
```

In [2]:
```python
data = pd.read_csv("breast-cancer.data",header=None)
data.columns = ['Class','age','menopause','tumor-size','inv-nodes','node-caps','deg-mali
```

In [3]:
```python
data
```

Out[3]:

	Class	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat
0	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
1	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	right	right_up	no
2	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	left_low	no
3	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	right	left_up	no
4	no-recurrence-events	40-49	premeno	0-4	0-2	no	2	right	right_low	no
...	...	...	...	...	...	...	...	...	...	...
281	recurrence-events	30-39	premeno	30-34	0-2	no	2	left	left_up	no
282	recurrence-events	30-39	premeno	20-24	0-2	no	3	left	left_up	yes
283	recurrence-events	60-69	ge40	20-24	0-2	no	1	right	left_up	no
284	recurrence-events	40-49	ge40	30-34	3-5	no	3	left	left_low	no
285	recurrence-events	50-59	ge40	30-34	3-5	no	3	left	left_low	no

286 rows × 10 columns

In [4]:
```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for i in data:
    data[i] = le.fit_transform(data[i])
```

In [5]:
```python
1  x= data.drop("irradiat",axis=1)
2  y = data["irradiat"]
```

In [6]:
```python
1  x
```

Out[6]:

	Class	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad
0	0	1	2	5	0	1	2	0	2
1	0	2	2	3	0	1	1	1	5
2	0	2	2	3	0	1	1	0	2
3	0	4	0	2	0	1	1	1	3
4	0	2	2	0	0	1	1	1	4
...	...	...	...	...	...	...	...	...	...
281	1	1	2	5	0	1	1	0	3
282	1	1	2	3	0	1	2	0	3
283	1	4	0	3	0	1	0	1	3
284	1	2	0	5	4	1	2	0	2
285	1	3	0	5	4	1	2	0	2

286 rows × 9 columns

In [7]:
```python
1  y
```

Out[7]:
```
0      0
1      0
2      0
3      0
4      0
      ..
281    0
282    1
283    0
284    0
285    0
Name: irradiat, Length: 286, dtype: int32
```

In [8]:
```python
1  x.shape
```

Out[8]: (286, 9)

In [9]:
```python
1  y.shape
```

Out[9]: (286,)

In [10]:
```python
1  from sklearn.model_selection import train_test_split
2  xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.25,random_state=0)
```

```
In [11]:  1  xtrain1 = np.array(xtrain)
          2  xtest1 = np.array(xtest)
          3  ytrain1 = np.array(ytrain)
          4  ytest1 = np.array(ytest)
```

```
In [12]:  1  xtrain1 = xtrain1.reshape(xtrain1.shape[0],xtrain1.shape[1],1)
          2  xtest1 = xtest1.reshape(xtest1.shape[0],xtest1.shape[1],1)
```

```
In [13]:  1  xtrain1.shape
```

Out[13]: (214, 9, 1)

```
In [14]:  1  xtest1.shape
```

Out[14]: (72, 9, 1)

```
In [15]:  1  ytrain1.shape
```

Out[15]: (214,)

```
In [16]:  1  ytest1.shape
```

Out[16]: (72,)

```
In [17]:  1  from keras.layers import LSTM,Dense,Activation,Flatten
          2  from keras.models import Sequential
```

```
In [18]:  1  model = Sequential()
          2  model.add(LSTM(100,input_shape=(xtrain1.shape[1],1)))
          3  model.add(Dense(10))
          4  model.add(Dense(8))
          5  model.add(Dense(1))
          6  model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100)	40800
dense (Dense)	(None, 10)	1010
dense_1 (Dense)	(None, 8)	88
dense_2 (Dense)	(None, 1)	9

Total params: 41,907
Trainable params: 41,907
Non-trainable params: 0

```
In [19]:  1  model.compile(metrics= ["accuracy"],optimizer="adam",loss="categorical_crossentropy")
```

In [20]:
```
1 history = model.fit(xtrain1, ytrain1,batch_size=10,epochs=250)
```

```
7617
Epoch 10/250
22/22 [==============================] - 0s 5ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 11/250
22/22 [==============================] - 0s 5ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 12/250
22/22 [==============================] - 0s 6ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 13/250
22/22 [==============================] - 0s 5ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 14/250
22/22 [==============================] - 0s 5ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 15/250
22/22 [==============================] - 0s 5ms/step - loss: 2.8410e-08 - accuracy: 0.
7617
Epoch 16/250
```

In [21]:
```
1 history = pd.DataFrame(history.history)
2 history
```

Out[21]:

| | loss | accuracy |
|---|---|---|
| 0 | 2.840969e-08 | 0.761682 |
| 1 | 2.840969e-08 | 0.761682 |
| 2 | 2.840969e-08 | 0.761682 |
| 3 | 2.840969e-08 | 0.761682 |
| 4 | 2.840969e-08 | 0.761682 |
| ... | ... | ... |
| 245 | 2.840969e-08 | 0.761682 |
| 246 | 2.840969e-08 | 0.761682 |
| 247 | 2.840969e-08 | 0.761682 |
| 248 | 2.840969e-08 | 0.761682 |
| 249 | 2.840969e-08 | 0.761682 |

250 rows × 2 columns

In [22]:
```python
1  history.plot(figsize=(15,10))
```

Out[22]: <AxesSubplot:>
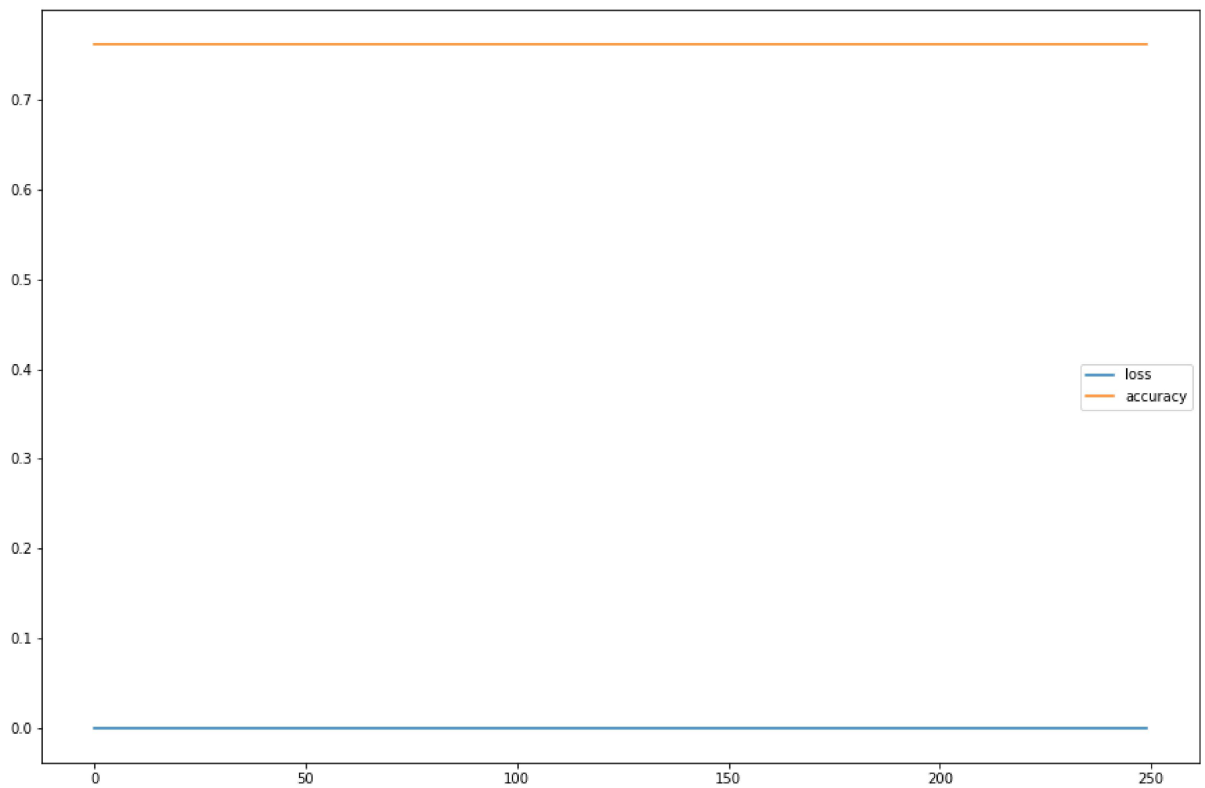
In [23]:
```python
1  ypredict = np.argmax(model.predict(xtest1), axis=-1)
```

In [24]:
```python
1  from sklearn.metrics import accuracy_score
2  accuracy_score(ytest1,ypredict)
```

Out[24]: 0.7638888888888888