In [1]:

```
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]:

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

In [3]:

```
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
x= data.drop("irradiat",axis=1)
y = data["irradiat"]
```

In [6]:

```python
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
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
x.shape
```

Out[8]:

```
(286, 9)
```

In [9]:

```
y.shape
```

Out[9]:

```
(286,)
```

In [10]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.25,random_state=0)
```

In [11]:

```
xtrain1 = np.array(xtrain)
xtest1 = np.array(xtest)
ytrain1 = np.array(ytrain)
ytest1 = np.array(ytest)
```

In [12]:

```
xtrain1 = xtrain1.reshape(xtrain1.shape[0],xtrain1.shape[1],1)
xtest1 = xtest1.reshape(xtest1.shape[0],xtest1.shape[1],1)
```

In [13]:

```
xtrain1.shape
```

Out[13]:

```
(214, 9, 1)
```

In [14]:

```
xtest1.shape
```

Out[14]:

```
(72, 9, 1)
```

In [15]:

```
ytrain1.shape
```

Out[15]:

```
(214,)
```

In [16]:

```
ytest1.shape
```

Out[16]:

```
(72,)
```

In [17]:

```
from keras.layers import LSTM,Dense,Activation,Flatten,Dropout
from keras.models import Sequential
```

In [18]:

```
model = Sequential()
model.add(LSTM(256,input_shape=(xtrain1.shape[1],1)))
model.add(Dropout(0.2))
model.add(Dense(1, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

In [19]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 256)               264192    
_________________________________________________________________
dropout (Dropout)            (None, 256)               0         
_________________________________________________________________
dense (Dense)                (None, 1)                 257       
=================================================================
Total params: 264,449
Trainable params: 264,449
Non-trainable params: 0
_________________________________________________________________
```

In [20]:

```
model.compile(metrics= ["accuracy"],optimizer="adam",loss="categorical_crossentropy"
```

In [21]:

```
history = model.fit(xtrain1, ytrain1,batch_size=10,epochs=250)
```

```
0.2383
Epoch 94/250
22/22 [==============================] - 0s 17ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 95/250
22/22 [==============================] - 0s 19ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 96/250
22/22 [==============================] - 0s 19ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 97/250
22/22 [==============================] - 0s 20ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 98/250
22/22 [==============================] - 0s 20ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 99/250
22/22 [==============================] - 0s 20ms/step - loss: 0.0000e+00 - accuracy:
0.2383
Epoch 100/250
```

In [22]:

```
history = pd.DataFrame(history.history)
history
```

Out[22]:

	loss	accuracy
0	0.0	0.238318
1	0.0	0.238318
2	0.0	0.238318
3	0.0	0.238318
4	0.0	0.238318
...	...	...
245	0.0	0.238318
246	0.0	0.238318
247	0.0	0.238318
248	0.0	0.238318
249	0.0	0.238318

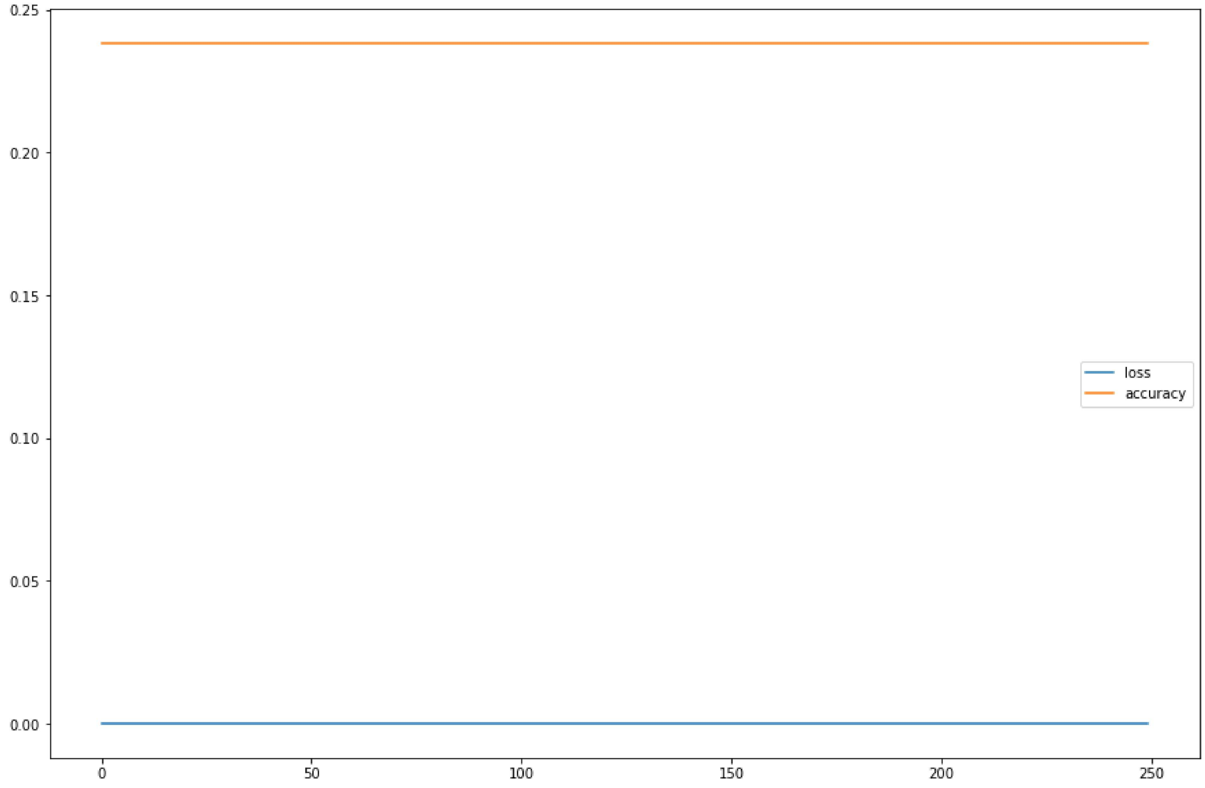
250 rows × 2 columns

In [23]:

```
history.plot(figsize=(15,10))
```

Out[23]:

```
<AxesSubplot:>
```



In [24]:

```
ypredict = np.argmax(model.predict(xtest1), axis=-1)
```

In [25]:

```
from sklearn.metrics import accuracy_score
accuracy_score(ytest1,ypredict)
```

Out[25]:

```
0.7638888888888888
```