In [36]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [37]:

```
data=pd.read_csv("Cancer_data_num.csv")
```

In [38]:

```
data.head()
```

Out[38]:

	Patient Id	Age	Gender	hairfall	fatigue	lump	weightloss	fever/nightsweats	skinchanges	m
0	P1	33	3	2	2	2	2	2	2	
1	P2	17	3	2	2	2	2	2	1	
2	P3	35	3	2	2	2	1	2	2	
3	P4	37	3	2	2	2	1	1	1	
4	P5	46	3	2	2	1	2	1	1	

In [39]:

```
data.tail()
```

Out[39]:

	Patient Id	Age	Gender	hairfall	fatigue	lump	weightloss	fever/nightsweats	skinchanges
6139	P6140	22	3	2	2	2	2	2	2
6140	P6141	42	0	2	2	2	2	2	2
6141	P6142	37	3	2	2	2	2	1	2
6142	P6143	35	0	2	2	2	1	2	2
6143	P6144	24	3	2	2	2	1	1	2

In [40]:

```
data['level']
```

Out[40]:

```
0       medium
1       medium
2         high
3       medium
4       medium
         ...  
6139      high
6140      high
6141    medium
6142      high
6143    medium
Name: level, Length: 6144, dtype: object
```

In [41]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6144 entries, 0 to 6143
Data columns (total 15 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   Patient Id                     6144 non-null   object
 1   Age                            6144 non-null   int64 
 2   Gender                         6144 non-null   int64 
 3   hairfall                       6144 non-null   int64 
 4   fatigue                        6144 non-null   int64 
 5   lump                           6144 non-null   int64 
 6   weightloss                     6144 non-null   int64 
 7   fever/nightsweats              6144 non-null   int64 
 8   skinchanges                    6144 non-null   int64 
 9   muscle/joint pain              6144 non-null   int64 
 10  bleeding/bruising              6144 non-null   int64 
 11  smoking                        6144 non-null   int64 
 12  alcoholuse                     6144 non-null   int64 
 13  indigestion/irregular bowel    6144 non-null   int64 
 14  level                          6144 non-null   object
dtypes: int64(13), object(2)
memory usage: 720.1+ KB
```

In [42]:

```
data.drop(['Patient Id'],axis=1,inplace=True)
```

In [43]:

```
data.drop(['Gender'],axis=1,inplace=True)
```

In [44]:

```
data.head()
```

Out[44]:

	Age	hairfall	fatigue	lump	weightloss	fever/nightsweats	skinchanges	muscle/joint pain	bleedi
0	33	2	2	2	2	2	2	1	
1	17	2	2	2	2	2	1	2	
2	35	2	2	2	1	2	2	2	
3	37	2	2	2	1	1	1	1	
4	46	2	2	1	2	1	1	2	

In [46]:

```
data['level'].replace('high','2',inplace=True)
data['level'].replace('medium','1',inplace=True)
data['level'].replace('low','0',inplace=True)
```

In [47]:

```
data.head()
```

Out[47]:

	Age	hairfall	fatigue	lump	weightloss	fever/nightsweats	skinchanges	muscle/joint pain	bleedi
0	33	2	2	2	2	2	2	1	
1	17	2	2	2	2	2	1	2	
2	35	2	2	2	1	2	2	2	
3	37	2	2	2	1	1	1	1	
4	46	2	2	1	2	1	1	2	

In [48]:

```
data.tail()
```

Out[48]:

	Age	hairfall	fatigue	lump	weightloss	fever/nightsweats	skinchanges	muscle/joint pain	ble
6139	22	2	2	2	2	2	2	1	
6140	42	2	2	2	2	2	2	1	
6141	37	2	2	2	2	1	2	1	
6142	35	2	2	2	1	2	2	2	
6143	24	2	2	2	1	1	2	2	

In [49]:

```
data.isnull().sum()
```

Out[49]:

```
Age                             0
hairfall                        0
fatigue                         0
lump                            0
weightloss                      0
fever/nightsweats               0
skinchanges                     0
muscle/joint pain               0
bleeding/bruising               0
smoking                         0
alcoholuse                      0
indigestion/irregular bowel     0
level                           0
dtype: int64
```

In [50]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6144 entries, 0 to 6143
Data columns (total 13 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   Age                           6144 non-null   int64 
 1   hairfall                      6144 non-null   int64 
 2   fatigue                       6144 non-null   int64 
 3   lump                          6144 non-null   int64 
 4   weightloss                    6144 non-null   int64 
 5   fever/nightsweats             6144 non-null   int64 
 6   skinchanges                   6144 non-null   int64 
 7   muscle/joint pain             6144 non-null   int64 
 8   bleeding/bruising             6144 non-null   int64 
 9   smoking                       6144 non-null   int64 
 10  alcoholuse                    6144 non-null   int64 
 11  indigestion/irregular bowel  6144 non-null   int64 
 12  level                         6144 non-null   object
dtypes: int64(12), object(1)
memory usage: 624.1+ KB
```

In [51]:

```
data.level
```

Out[51]:

```
0       1
1       1
2       2
3       1
4       1
       ..
6139    2
6140    2
6141    1
6142    2
6143    1
Name: level, Length: 6144, dtype: object
```

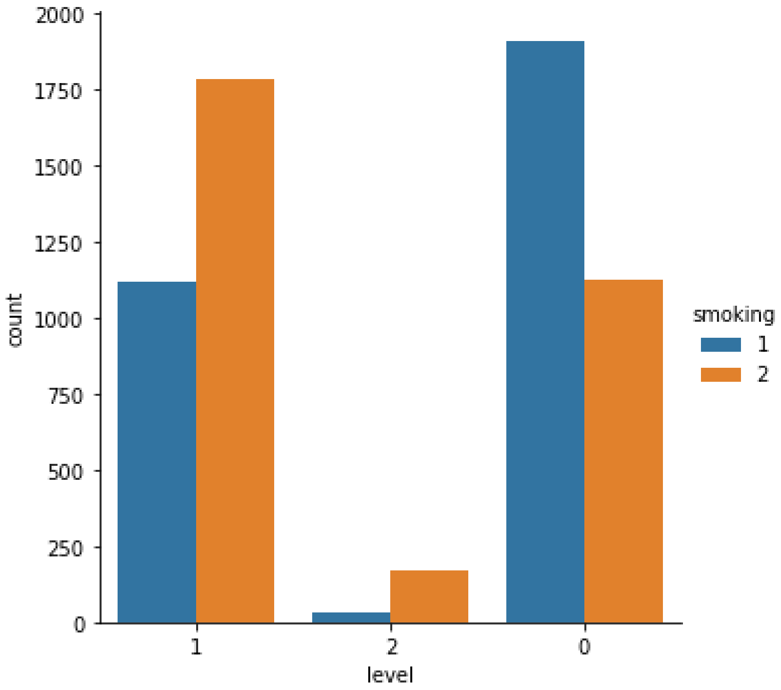
In [62]:

```
# import seaborn as sb
sb.factorplot('level',data=data,hue='smoking',kind='count')
```

```
C:\Users\reddy\anaconda3\lib\site-packages\seaborn\categorical.py:3717: User
Warning: The `factorplot` function has been renamed to `catplot`. The origin
al name will be removed in a future release. Please update your code. Note t
hat the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in
`catplot`.
  warnings.warn(msg)
C:\Users\reddy\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x246219dbdc0>
```



RANDOM FOREST

In [63]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
X = data.drop('level',axis = 1)
y = data['level']
X_train,X_test,y_train,y_test=train_test_split(X,y)
```

In [64]:

```
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier()
model.fit(X_train,y_train)
```

Out[64]:

```
RandomForestClassifier()
```

Checking accuracy of Random Forest

In [65]:

```
model_score=model.score(X_test,y_test)
y_pred_randomF = model.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test, y_pred_randomF)*100)
print("Our model score is",model_score)
```

```
Accuracy score :  97.52604166666666
Our model score is 0.9752604166666666
```

In [67]:

```
print(y_pred_randomF)
```

```
['1' '1' '0' ... '1' '0' '0']
```

confusion matrix

In [69]:

```
cn=confusion_matrix(y_test,y_pred_randomF)
cn
```

Out[69]:

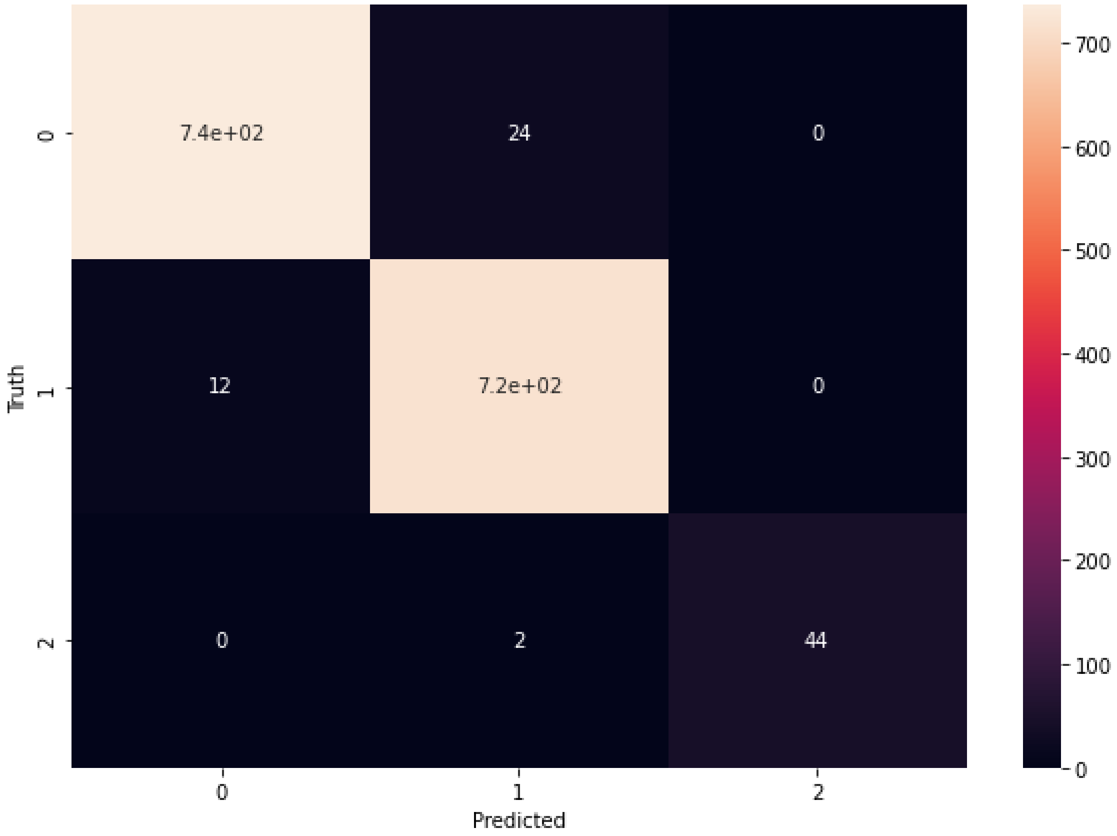
```
array([[736,  24,   0],
       [ 12, 718,   0],
       [  0,   2,  44]], dtype=int64)
```

In [70]:

```
import seaborn as sb
plt.figure(figsize=(10,7))
sb.heatmap(cn,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[70]:

```
Text(69.0, 0.5, 'Truth')
```



KMeans

In [71]:

```
from sklearn.cluster import KMeans
clf = KMeans()
clf.fit(X_train)
maxx = clf.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test,maxx)*100)
```

```
Accuracy score :  0.0
```

DecisionTreeClassifier

In [78]:

```
from sklearn.tree import DecisionTreeClassifier
tree_ = DecisionTreeClassifier()
tree_.fit(X_train,y_train)
y_pred = tree_.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test, y_pred)*100-17)
```

```
Accuracy score :  79.484375
```

In [ ]: