Project By : PRASAD JADHAV

In [1]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from imblearn.over_sampling import SMOTE

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import svm

from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
dataset = pd.read_csv('lung_cancer.csv')
dataset.shape
```

Out[3]:

```
(309, 16)
```

In [4]:

```
dataset.head()
```

Out[4]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLER
0	M	69	1	2	2	1	1	2	
1	M	74	2	1	1	1	2	2	
2	F	59	1	1	1	2	1	2	
3	M	63	2	2	2	1	1	1	
4	F	63	1	2	1	1	1	1	

In [5]:

```
dataset.tail()
```

Out[5]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALL
304	F	56	1	1	1	2	2	2	
305	M	70	2	1	1	1	1	2	
306	M	58	2	1	1	1	1	1	
307	M	67	2	1	2	1	1	2	
308	M	62	1	1	1	2	1	2	

In [6]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   GENDER                 309 non-null    object
 1   AGE                    309 non-null    int64 
 2   SMOKING                309 non-null    int64 
 3   YELLOW_FINGERS         309 non-null    int64 
 4   ANXIETY                309 non-null    int64 
 5   PEER_PRESSURE          309 non-null    int64 
 6   CHRONIC DISEASE        309 non-null    int64 
 7   FATIGUE                309 non-null    int64 
 8   ALLERGY                309 non-null    int64 
 9   WHEEZING               309 non-null    int64 
 10  ALCOHOL CONSUMING      309 non-null    int64 
 11  COUGHING               309 non-null    int64 
 12  SHORTNESS OF BREATH    309 non-null    int64 
 13  SWALLOWING DIFFICULTY  309 non-null    int64 
 14  CHEST PAIN             309 non-null    int64 
 15  LUNG_CANCER            309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```

In [7]:

```
dataset.isnull().sum()
```

Out[7]:

```
GENDER                   0
AGE                      0
SMOKING                  0
YELLOW_FINGERS           0
ANXIETY                  0
PEER_PRESSURE            0
CHRONIC DISEASE          0
FATIGUE                  0
ALLERGY                  0
WHEEZING                 0
ALCOHOL CONSUMING        0
COUGHING                 0
SHORTNESS OF BREATH      0
SWALLOWING DIFFICULTY    0
CHEST PAIN               0
LUNG_CANCER              0
dtype: int64
```

In [8]:

```
dataset.isna().sum()
```

Out[8]:

```
GENDER                   0
AGE                      0
SMOKING                  0
YELLOW_FINGERS           0
ANXIETY                  0
PEER_PRESSURE            0
CHRONIC DISEASE          0
FATIGUE                  0
ALLERGY                  0
WHEEZING                 0
ALCOHOL CONSUMING        0
COUGHING                 0
SHORTNESS OF BREATH      0
SWALLOWING DIFFICULTY    0
CHEST PAIN               0
LUNG_CANCER              0
dtype: int64
```

In [9]:

```
dataset.duplicated().sum()
```

Out[9]:

```
33
```

In [10]:

```
dataset = dataset.drop_duplicates()
```

In [11]:

```
dataset.describe()
```

Out[11]:

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIG
count	276.000000	276.000000	276.000000	276.000000	276.000000	276.000000	276.0000
mean	62.909420	1.543478	1.576087	1.496377	1.507246	1.521739	1.6630
std	8.379355	0.499011	0.495075	0.500895	0.500856	0.500435	0.4735
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0000
25%	57.750000	1.000000	1.000000	1.000000	1.000000	1.000000	1.0000
50%	62.500000	2.000000	2.000000	1.000000	2.000000	2.000000	2.0000
75%	69.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.0000
max	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.0000

In [12]:
```
dataset.corr()
```

Out[12]:

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FA
AGE	1.000000	-0.073410	0.025773	0.050605	0.037848	-0.003431	0.0
SMOKING	-0.073410	1.000000	-0.020799	0.153389	-0.030364	-0.149415	-0.0
YELLOW_FINGERS	0.025773	-0.020799	1.000000	0.558344	0.313067	0.015316	-0.0
ANXIETY	0.050605	0.153389	0.558344	1.000000	0.210278	-0.006938	-0.1
PEER_PRESSURE	0.037848	-0.030364	0.313067	0.210278	1.000000	0.042893	0.0
CHRONIC DISEASE	-0.003431	-0.149415	0.015316	-0.006938	0.042893	1.000000	-0.0
FATIGUE	0.021606	-0.037803	-0.099644	-0.181474	0.094661	-0.099411	1.0
ALLERGY	0.037139	-0.030179	-0.147130	-0.159451	-0.066887	0.134309	-0.0
WHEEZING	0.052803	-0.147081	-0.058756	-0.174009	-0.037769	-0.040546	0.1
ALCOHOL CONSUMING	0.052049	-0.052771	-0.273643	-0.152228	-0.132603	0.010144	-0.1
COUGHING	0.168654	-0.138553	0.020803	-0.218843	-0.068224	-0.160813	0.1
SHORTNESS OF BREATH	-0.009189	0.051761	-0.109959	-0.155678	-0.214115	-0.011760	0.4
SWALLOWING DIFFICULTY	0.003199	0.042152	0.333349	0.478820	0.327764	0.068263	-0.1
CHEST PAIN	-0.035806	0.106984	-0.099169	-0.123182	-0.074655	-0.048895	0.0

In [13]:
```
dataset.cov()
```

Out[13]:

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FA
AGE	70.213584	-0.306957	0.106917	0.212398	0.158841	-0.014387	0.0
SMOKING	-0.306957	0.249012	-0.005138	0.038340	-0.007589	-0.037312	-0.0
YELLOW_FINGERS	0.106917	-0.005138	0.245099	0.138458	0.077628	0.003794	-0.0
ANXIETY	0.212398	0.038340	0.138458	0.250896	0.052754	-0.001739	-0.0
PEER_PRESSURE	0.158841	-0.007589	0.077628	0.052754	0.250856	0.010751	0.0
CHRONIC DISEASE	-0.014387	-0.037312	0.003794	-0.001739	0.010751	0.250435	-0.0
FATIGUE	0.085731	-0.008933	-0.023360	-0.043043	0.022451	-0.023557	0.2
ALLERGY	0.155191	-0.007510	-0.036324	-0.039829	-0.016706	0.033518	-0.0
WHEEZING	0.220646	-0.036601	-0.014506	-0.043465	-0.009433	-0.010119	0.0
ALCOHOL CONSUMING	0.217339	-0.013123	-0.067510	-0.037997	-0.033096	0.002530	-0.0
COUGHING	0.699644	-0.034229	0.005099	-0.054269	-0.016917	-0.039842	0.0
SHORTNESS OF BREATH	-0.037233	0.012490	-0.026324	-0.037708	-0.051858	-0.002846	0.0
SWALLOWING DIFFICULTY	0.013399	0.010514	0.082490	0.119881	0.082055	0.017075	-0.0
CHEST PAIN	-0.149275	0.026561	-0.024427	-0.030698	-0.018603	-0.012174	0.0

In [14]:

```
dataset['GENDER'] = dataset['GENDER'].map({'M':1,'F':0})
dataset['LUNG_CANCER'] = dataset['LUNG_CANCER'].map({'YES':1,'NO':0})
```

In [15]:

```
dataset.sample(11)
```

Out[15]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALL
194	1	63	1	1	1	1	2	2	
234	1	77	1	2	1	2	1	2	
176	0	70	1	2	1	1	2	2	
253	0	67	2	2	2	2	1	2	
59	1	69	2	2	2	2	1	2	
21	0	64	1	2	2	2	1	1	
188	1	65	2	2	2	2	2	1	
102	1	64	2	1	1	1	1	2	
205	1	62	1	2	2	2	1	2	
180	1	63	2	2	2	2	1	1	
227	1	71	1	2	2	1	2	1	

In [16]:

```
dataset.nunique()
```

Out[16]:

```
GENDER                     2
AGE                       39
SMOKING                    2
YELLOW_FINGERS             2
ANXIETY                    2
PEER_PRESSURE              2
CHRONIC DISEASE            2
FATIGUE                    2
ALLERGY                    2
WHEEZING                   2
ALCOHOL CONSUMING          2
COUGHING                   2
SHORTNESS OF BREATH        2
SWALLOWING DIFFICULTY      2
CHEST PAIN                 2
LUNG_CANCER                2
dtype: int64
```

In [17]:

```
dataset.dtypes
```

Out[17]:

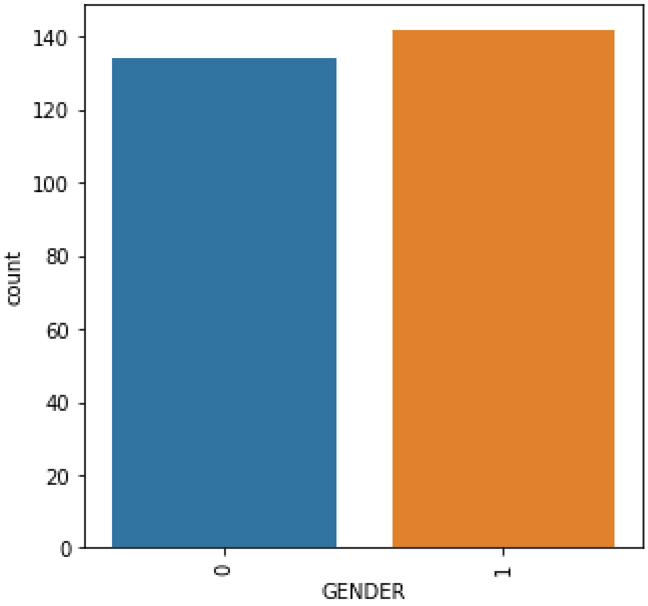
```
GENDER                   int64
AGE                      int64
SMOKING                  int64
YELLOW_FINGERS           int64
ANXIETY                  int64
PEER_PRESSURE            int64
CHRONIC DISEASE          int64
FATIGUE                  int64
ALLERGY                  int64
WHEEZING                 int64
ALCOHOL CONSUMING        int64
COUGHING                 int64
SHORTNESS OF BREATH      int64
SWALLOWING DIFFICULTY    int64
CHEST PAIN               int64
LUNG_CANCER              int64
dtype: object
```

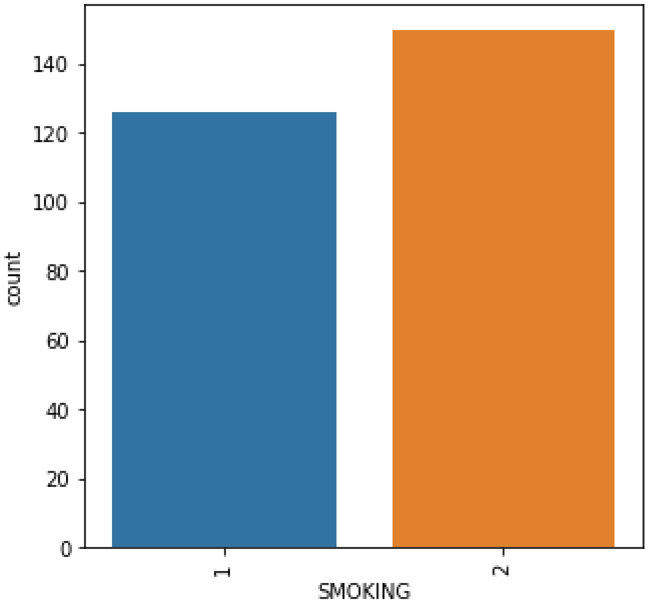
In [18]:

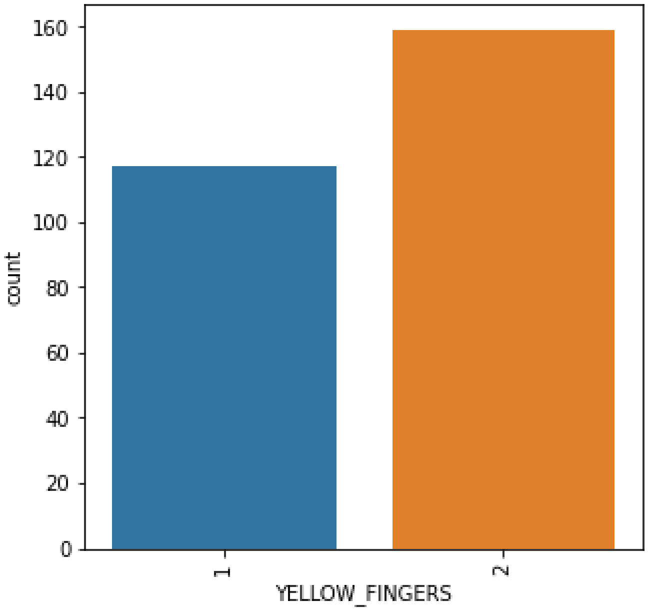
```
df = dataset[['GENDER', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY',
       'PEER_PRESSURE', 'CHRONIC DISEASE', 'FATIGUE ', 'ALLERGY ', 'WHEEZING',
       'ALCOHOL CONSUMING', 'COUGHING', 'SHORTNESS OF BREATH',
       'SWALLOWING DIFFICULTY', 'CHEST PAIN', 'LUNG_CANCER']]
```

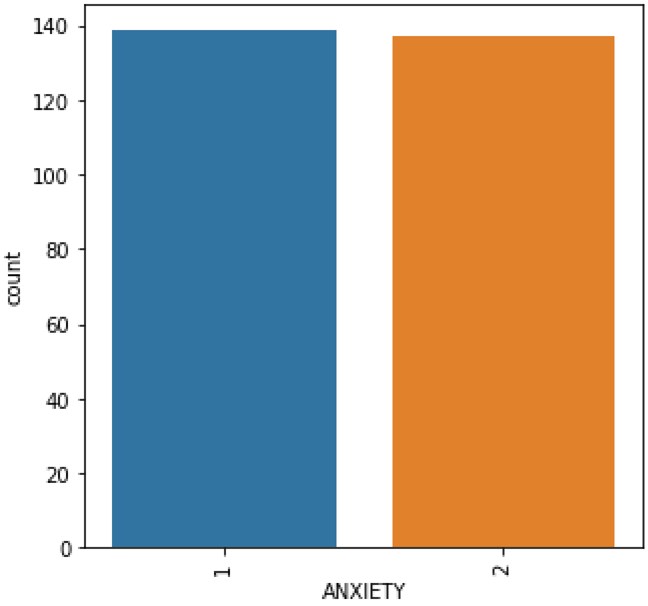
In [19]:

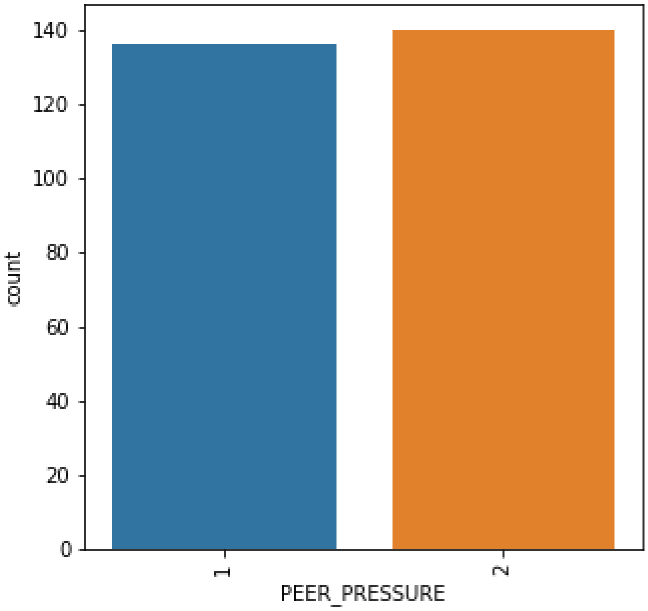
```
for i in df.columns:
    plt.figure(figsize=(5,5))
    sns.countplot(df[i], data = df)
    plt.xticks(rotation = 90)
    plt.show()
```

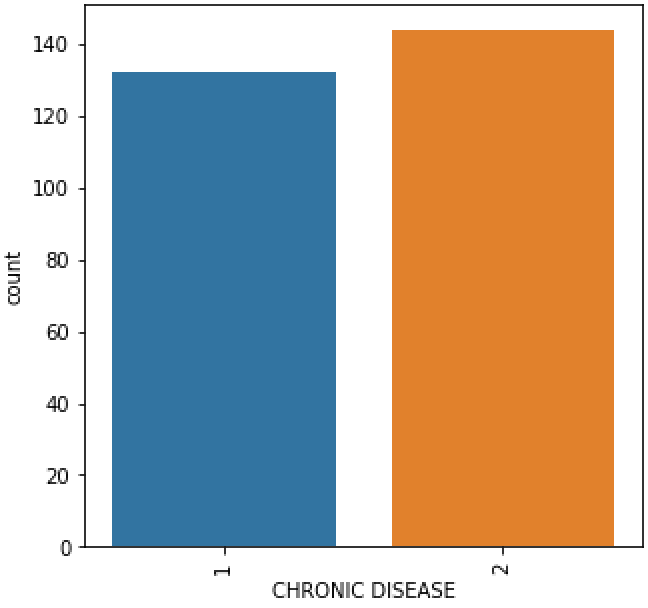

count
0
20
40
60
80
100
120
140
0
1
GENDER

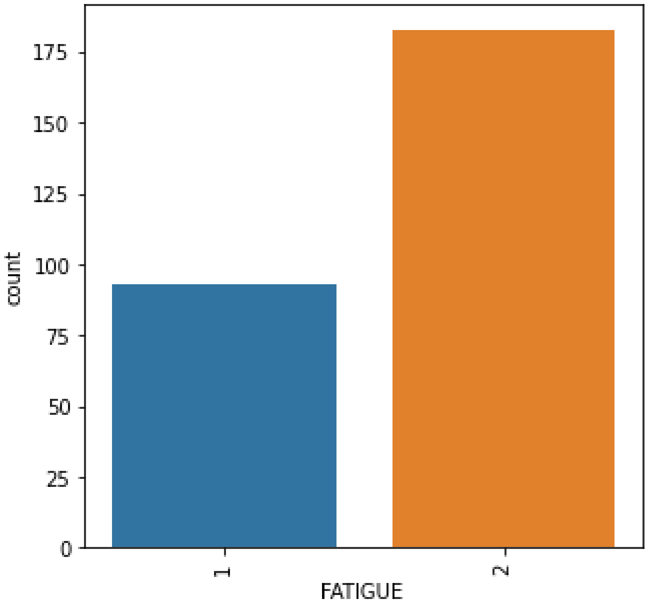

count
0
20
40
60
80
100
120
140
1
2
SMOKING

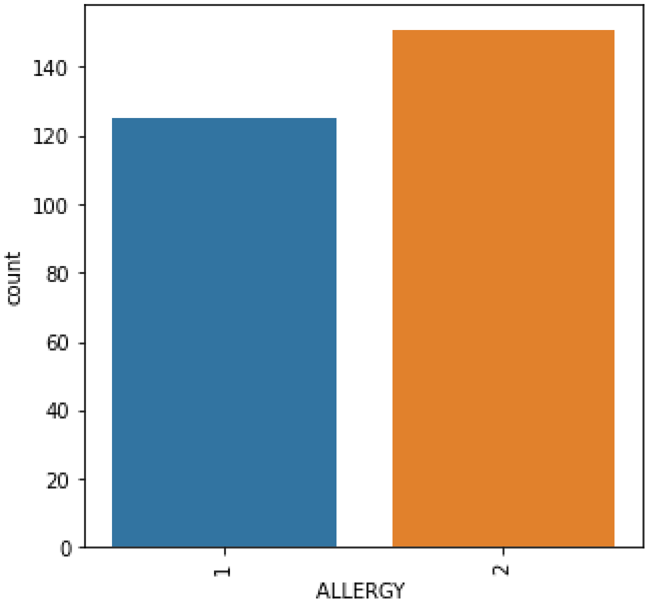

count
0
20
40
60
80
100
120
140
160
1
2
YELLOW_FINGERS

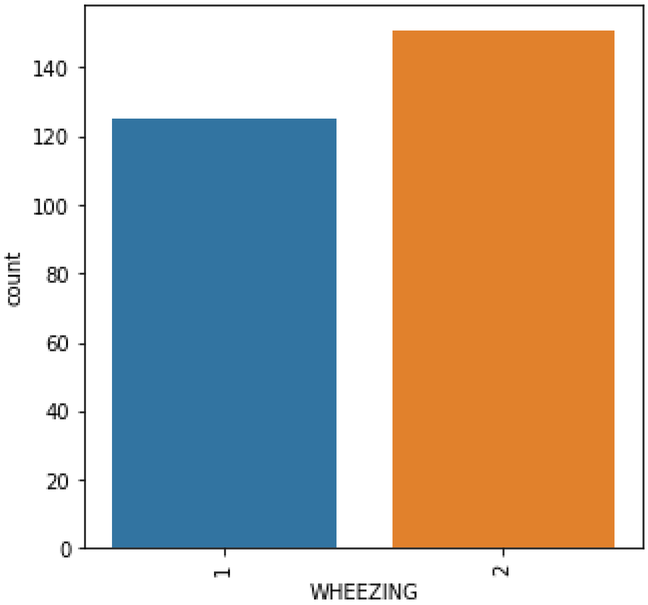

140
120
100
80
60
40
20
0
count
1
2
ANXIETY

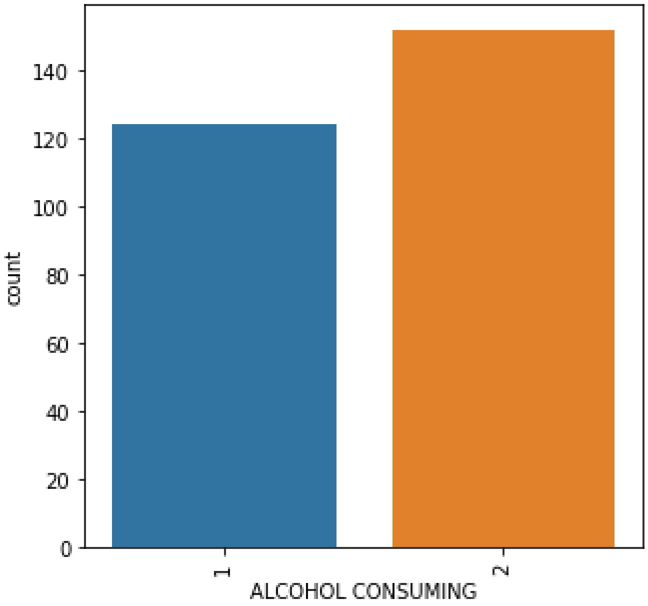

140
120
100
80
60
40
20
0
count
1
2
PEER_PRESSURE

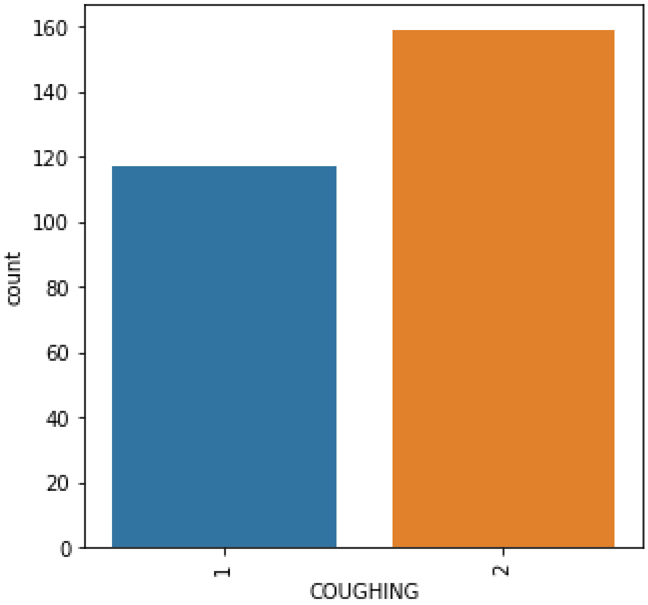

140
120
100
80
60
40
20
0
count
1
2
CHRONIC DISEASE

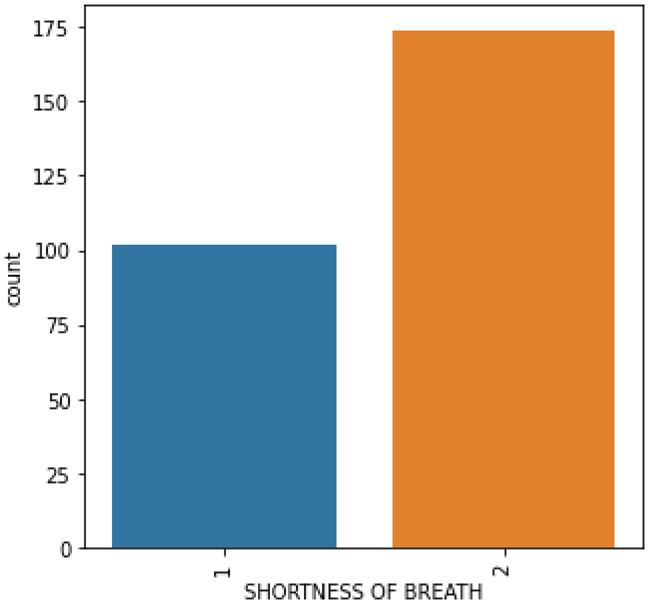

count
0
25
50
75
100
125
150
175
1
2
FATIGUE

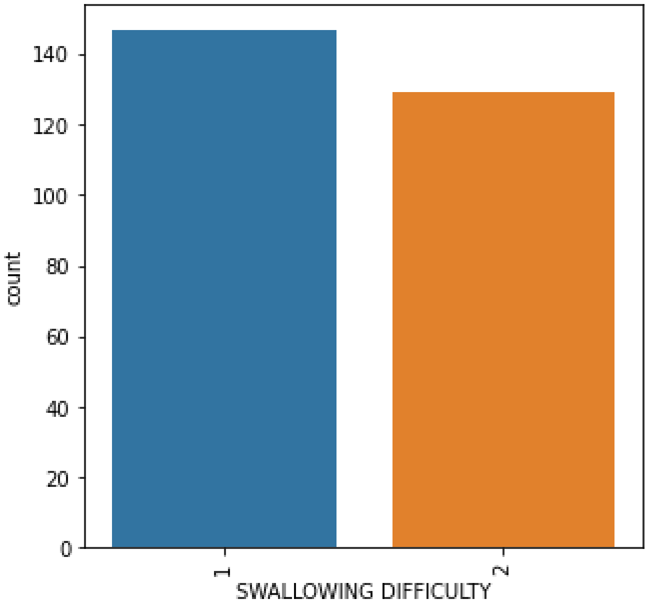

count
0
20
40
60
80
100
120
140
1
2
ALLERGY

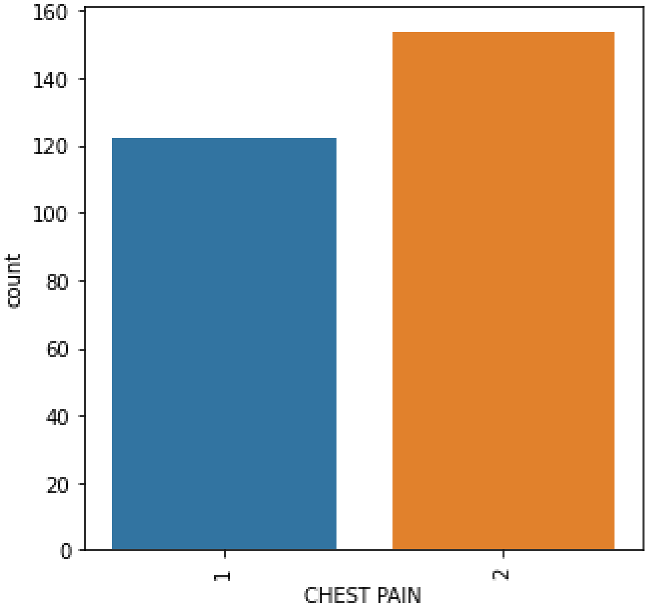

count
0
20
40
60
80
100
120
140
1
2
WHEEZING

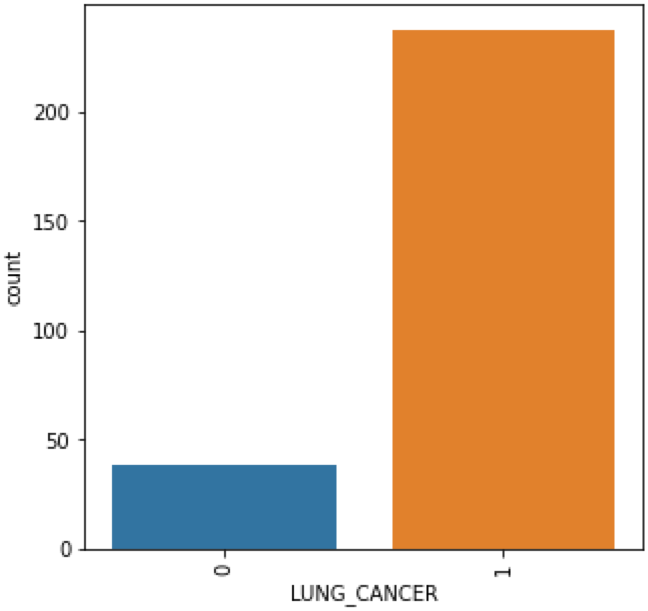

count
0
20
40
60
80
100
120
140
1
2
ALCOHOL CONSUMING


count
0
20
40
60
80
100
120
140
160
1
2
COUGHING


count
0
25
50
75
100
125
150
175
1
2
SHORTNESS OF BREATH


count
0
20
40
60
80
100
120
140
1
2
SWALLOWING DIFFICULTY


count
0
20
40
60
80
100
120
140
160
1
2
CHEST PAIN


count
0
50
100
150
200
0
1
LUNG_CANCER

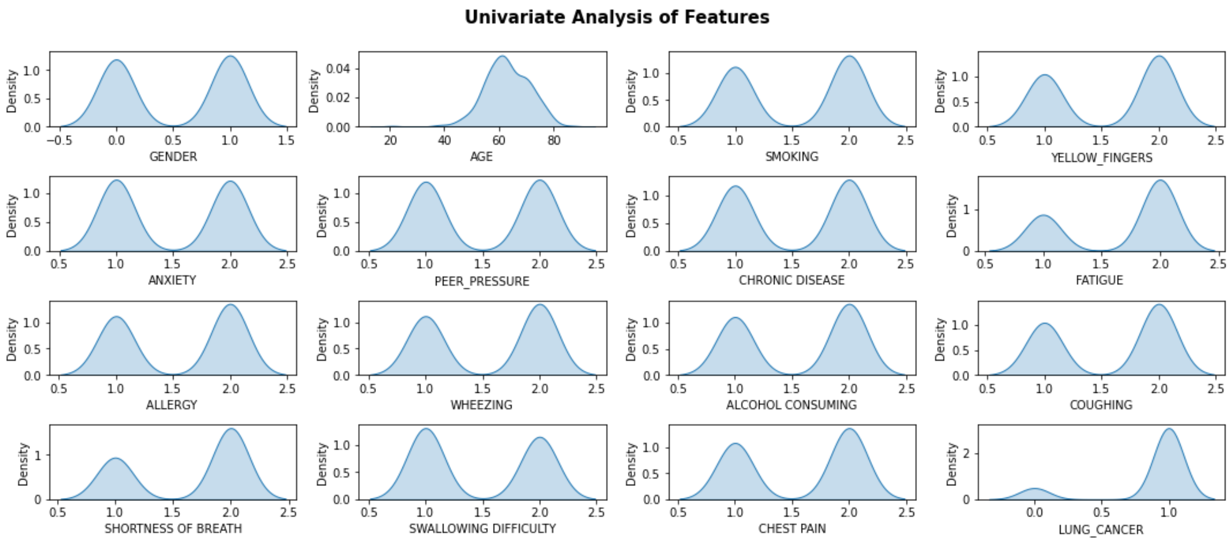
In [20]:

```
features = [feature for feature in dataset.columns]
```

In [21]:

```
plt.figure(figsize=(15,15))
plt.suptitle('Univariate Analysis of Features',fontweight='bold',fontsize=15,y

for i in range(0,len(features)):
    plt.subplot(10,4,i+1)
    sns.kdeplot(x=dataset[features[i]],shade=True)
    plt.tight_layout()
```



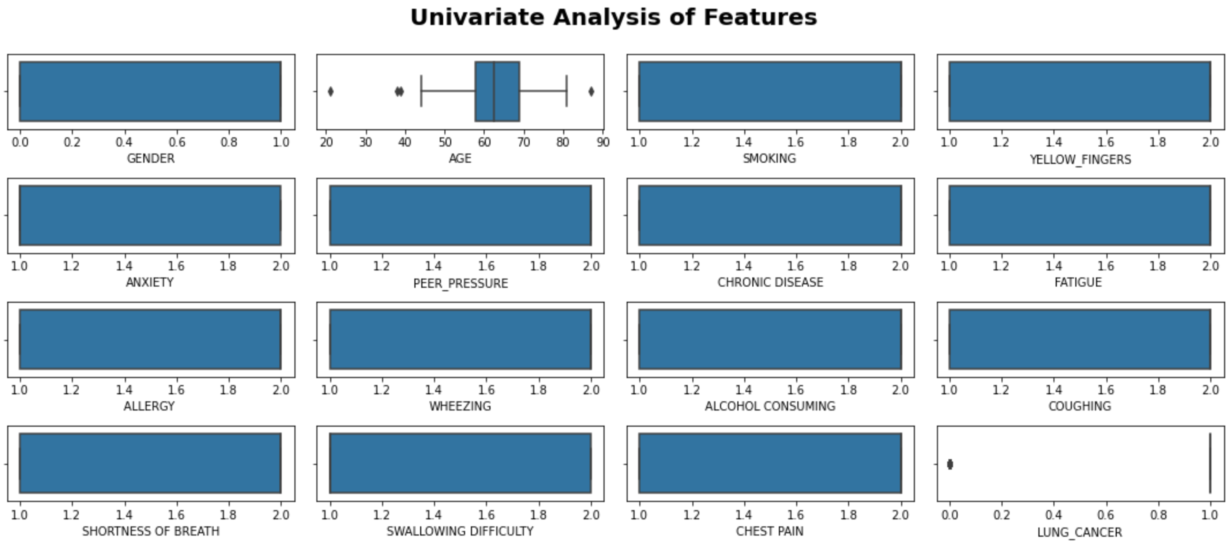
In [22]:

```
num_features = [feature for feature in dataset.columns]
```

In [23]:

```
plt.figure(figsize = (15,15))
plt.suptitle('Univariate Analysis of Features',fontweight='bold',fontsize=20,y

for i in range(0,len(num_features)):
    plt.subplot(10,4,i+1)
    sns.boxplot(data=dataset,x=features[i])
    plt.xlabel(num_features[i])
    plt.tight_layout()
```



In [ ]:

```
def remove_outliers(in_dataset, in_cols):
    first_quartile = in_dataset[in_cols].quantile(0.25)
    third_quartile = in_dataset[in_cols].quantile(0.75)
    iqr = third_quartile - first_quartile
    upper_limit = third_quartile + 1.5 * iqr
    lower_limit = first_quartile - 1.5 * iqr
    in_dataset.loc[(in_dataset[in_cols] > upper_limit), in_cols] = upper_limit
    in_dataset.loc[(in_dataset[in_cols] < lower_limit), in_cols] = lower_limit
    return in_dataset
```

In [ ]:

```
for features in num_features:
    dataset = remove_outliers(dataset,features)
```

In [ ]:

```
plt.figure(figsize = (20,250))
plt.suptitle('Univariate Analysis of Num Features',fontweight='bold',fontsize=

for i in range(0,len(num_features)):
    plt.subplot(85,3,i+1)
    sns.boxplot(data=dataset,x=num_features[i])
    plt.xlabel(num_features[i])
    plt.tight_layout()
```

In [24]:

```
dataset['LUNG_CANCER'].value_counts()
```

Out[24]:

```
1    238
0     38
Name: LUNG_CANCER, dtype: int64
```

In [26]:

```
X = dataset.drop('LUNG_CANCER',axis=1)
y = dataset['LUNG_CANCER']
```

In [27]:

```
X_res,y_res = SMOTE().fit_resample(X,y)
```

In [28]:

```
X_train,X_test,y_train,y_test = train_test_split(X_res,y_res,test_size=0.20,ra
```

In [29]:

```
st = StandardScaler()
X_train = st.fit_transform(X_train)
X_test = st.fit_transform(X_test)
```

In [30]:

```
model_df = {}

def model_val(model,X,y):
    X_train,X_test,y_train,y_test = train_test_split(X_res,y_res,test_size=0.2
    model.fit(X_train,y_train)
    y_pred = model.predict(X_test)
    print(f'{model} Accuracy is {accuracy_score(y_test,y_pred)}')

    score = cross_val_score(model,X,y,cv=5,n_jobs=-1)
    print(f'{model} Average cross val score is {np.mean(score)}')
    model_df[model] = round(np.mean(score)*100,2)
```

In [31]:

```
model = LogisticRegression()
model_val(model,X,y)
```

```
LogisticRegression() Accuracy is 0.96875
LogisticRegression() Average cross val score is 0.9057792207792208
```

In [32]:
```
model = DecisionTreeClassifier()
model_val(model,X,y)
```

```
DecisionTreeClassifier() Accuracy is 0.9479166666666666
DecisionTreeClassifier() Average cross val score is 0.8477922077922078
```

In [33]:
```
model = RandomForestClassifier()
model_val(model,X,y)
```

```
RandomForestClassifier() Accuracy is 0.96875
RandomForestClassifier() Average cross val score is 0.8949350649350649
```

In [34]:
```
model = GradientBoostingClassifier()
model_val(model,X,y)
```

```
GradientBoostingClassifier() Accuracy is 0.9375
GradientBoostingClassifier() Average cross val score is 0.8841558441558443
```

In [35]:
```
model_df
```

Out[35]:
```
{LogisticRegression(): 90.58,
 DecisionTreeClassifier(): 84.78,
 RandomForestClassifier(): 89.49,
 GradientBoostingClassifier(): 88.42}
```

In [38]:
```
lr = LogisticRegression()
lr.fit(X_train,y_train)
```

Out[38]:
```
▾ LogisticRegression
LogisticRegression()
```

In [39]:
```
y_pred = lr.predict(X_test)
```

In [40]:
```
accuracy_score(y_test,y_pred)
```

Out[40]:
```
0.96875
```

In [41]:
```
import pickle
import joblib
```

In [42]:
```
pickle.dump(lr,open('lung_cancer_prediction.pkl','wb'))
```

In [43]:
```
model = pickle.load(open('lung_cancer_prediction.pkl','rb'))
```

In [44]:
```
dataset.head()
```

Out[44]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLER
0	1	69	1	2	2	1	1	2	
1	1	74	2	1	1	1	2	2	
2	0	59	1	1	1	2	1	2	
3	1	63	2	2	2	1	1	1	
4	0	63	1	2	1	1	1	1	

In [46]:

```
new_df = pd.DataFrame({
    'GENDER':1,
    'AGE':69,
    'SMOKING':1,
    'YELLOW_FINGERS':2,
    'ANXIETY':2,
    'PEER_PRESSURE':1,
    'CHRONIC DISEASE':1,
    'FATIGUE':2,
    'ALLERGY':1,
    'WHEEZING':2,
    'ALCOHOL CONSUMING':2,
    'COUGHING':2,
    'SHORTNESS OF BREATH':2,
    'SWALLOWING DIFFICULTY':2,
    'CHEST PAIN':2
},index=[0])
```

In [47]:

```
model.predict(new_df)
```

Out[47]:

```
array([1], dtype=int64)
```

In [48]:

```
p = model.predict(new_df)

prob = model.predict_proba(new_df)
if p == 1:
    print('Lung Cancer!')
    print(f'You will be Lung Cancer! with Probability of {prob[0][1]:.2f}')
else:
    print('Not-Lung Cancer!')
```

```
Lung Cancer!
You will be Lung Cancer! with Probability of 1.00
```

CONNECT WITH ME :

LinkedIn GitHub kaggle Medium

PRASADMJADHAV2