Sir, I changeded the Data set because i did not find it suitable for Regression

In [40]:
```
import pandas as pd
import numpy as np
%matplotlib inline
from numpy import arange
from matplotlib import pyplot as plt
from scipy.stats import norm
from pandas_profiling import ProfileReport
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Lung Cancer Status :

0 -- > Yes

1 -- > NO

DataFraming

Read .csv file into pandas

In [21]:
```
data  = pd.read_csv('survey lung cancer.csv')
data.head()
```

Out[21]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTI OF BRI
0	M	69	1	2	2	1	1	2	1	2	2	2	
1	M	74	2	1	1	1	2	2	2	1	1	1	
2	F	59	1	1	1	2	1	2	1	2	1	2	
3	M	63	2	2	2	1	1	1	1	1	2	1	
4	F	63	1	2	1	1	1	1	1	2	1	2	

Exploratory data analysis

In [22]:
```
#Shape of data
print(data.shape)
#dtypes of data
print(data.dtypes)
```

```
(309, 16)
GENDER                  object
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             object
dtype: object
```

In [23]:
```
# Info of data
data.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 [42]:
```
data.memory_usage()
```

Out[42]:
```
Index                    128
GENDER                  2472
AGE                     2472
SMOKING                 2472
YELLOW_FINGERS          2472
ANXIETY                 2472
PEER_PRESSURE           2472
CHRONIC DISEASE         2472
FATIGUE                 2472
ALLERGY                 2472
WHEEZING                2472
ALCOHOL CONSUMING       2472
COUGHING                2472
SHORTNESS OF BREATH     2472
SWALLOWING DIFFICULTY   2472
CHEST PAIN              2472
LUNG_CANCER             2472
dtype: int64
```

In [43]:
```
data.memory_usage().sum()
```

Out[43]:
```
39680
```

In [44]:
```
data.describe()
```

Out[44]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000
mean	0.475728	62.673139	1.563107	1.569579	1.498382	1.501618	1.504854	1.673139	1.556634	1.556634	1.556634
std	0.500221	8.210301	0.496806	0.495938	0.500808	0.500808	0.500787	0.469827	0.497588	0.497588	0.497588
min	0.000000	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	0.000000	57.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING
50%	0.000000	62.000000	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
75%	1.000000	69.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	1.000000	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

In [45]:
```
data.mean()
```

Out[45]:
```
GENDER                     0.475728
AGE                       62.673139
SMOKING                    1.563107
YELLOW_FINGERS             1.569579
ANXIETY                    1.498382
PEER_PRESSURE              1.501618
CHRONIC DISEASE            1.504854
FATIGUE                    1.673139
ALLERGY                    1.556634
WHEEZING                   1.556634
ALCOHOL CONSUMING          1.556634
COUGHING                   1.579288
SHORTNESS OF BREATH        1.640777
SWALLOWING DIFFICULTY      1.469256
CHEST PAIN                 1.556634
LUNG_CANCER                0.126214
dtype: float64
```

In [46]:
```
data.var()
```

Out[46]:
```
GENDER                     0.250221
AGE                       67.409049
SMOKING                    0.246816
YELLOW_FINGERS             0.245955
ANXIETY                    0.250809
PEER_PRESSURE              0.250809
CHRONIC DISEASE            0.250788
FATIGUE                    0.220737
ALLERGY                    0.247594
WHEEZING                   0.247594
ALCOHOL CONSUMING          0.247594
COUGHING                   0.244505
SHORTNESS OF BREATH        0.230929
```

```
SWALLOWING DIFFICULTY     0.249863
CHEST PAIN                0.247594
LUNG_CANCER               0.110642
dtype: float64
```

In [48]:
```
data.skew()
```

Out[48]:
```
GENDER                    0.097677
AGE                      -0.395086
SMOKING                  -0.255705
YELLOW_FINGERS           -0.282425
ANXIETY                   0.006504
PEER_PRESSURE            -0.006504
CHRONIC DISEASE          -0.019513
FATIGUE                  -0.741836
ALLERGY                  -0.229118
WHEEZING                 -0.229118
ALCOHOL CONSUMING        -0.229118
COUGHING                 -0.322786
SHORTNESS OF BREATH      -0.589714
SWALLOWING DIFFICULTY     0.123812
CHEST PAIN               -0.229118
LUNG_CANCER               2.262112
dtype: float64
```

In [49]:
```
data.kurtosis()
```

Out[49]:
```
GENDER                   -2.003469
AGE                       1.746558
SMOKING                  -1.947261
YELLOW_FINGERS           -1.932789
ANXIETY                  -2.013029
PEER_PRESSURE            -2.013029
CHRONIC DISEASE          -2.012689
FATIGUE                  -1.459167
ALLERGY                  -1.960235
WHEEZING                 -1.960235
ALCOHOL CONSUMING        -1.960235
COUGHING                 -1.908203
SHORTNESS OF BREATH      -1.663044
SWALLOWING DIFFICULTY    -1.997643
CHEST PAIN               -1.960235
LUNG_CANCER               3.137414
dtype: float64
```

In [50]:
```
data.max()
```

Out[50]:
```
GENDER                   1
AGE                     87
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              1
dtype: int64
```

In [51]:
```
data.min()
```

Out[51]:
```
GENDER                   0
AGE                     21
SMOKING                  1
YELLOW_FINGERS           1
ANXIETY                  1
PEER_PRESSURE            1
CHRONIC DISEASE          1
FATIGUE                  1
ALLERGY                  1
WHEEZING                 1
ALCOHOL CONSUMING        1
COUGHING                 1
SHORTNESS OF BREATH      1
SWALLOWING DIFFICULTY    1
CHEST PAIN               1
LUNG_CANCER              0
dtype: int64
```

In [52]:
```
data.corr()
```

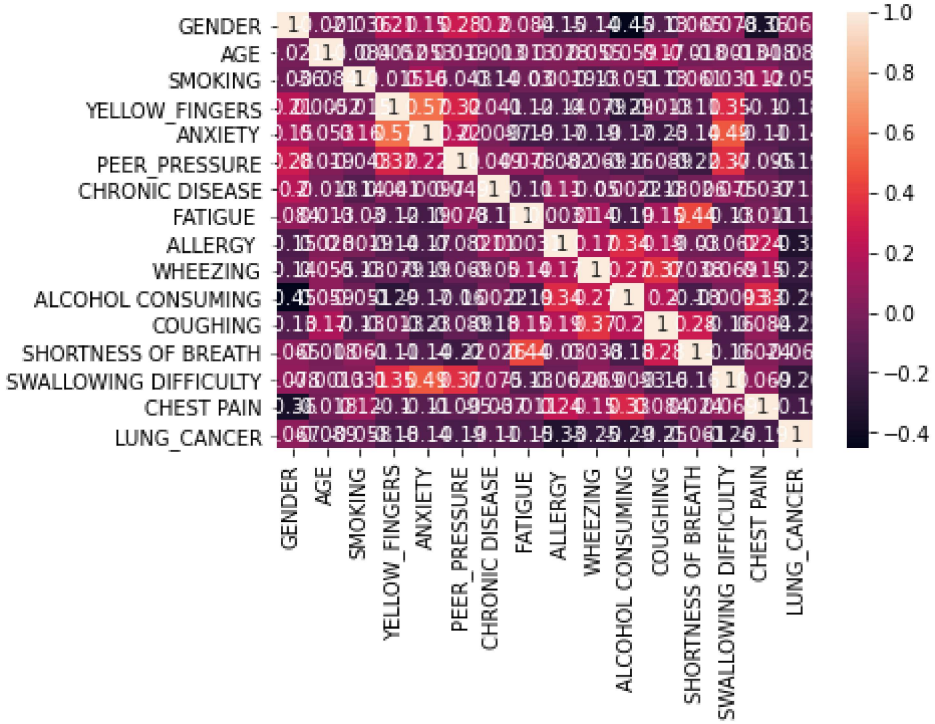
Out[52]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHO CONSUMIN
GENDER	1.000000	-0.021306	-0.036277	0.212959	0.152127	0.275564	0.204606	0.083560	-0.154251	-0.141207	-0.45426
AGE	-0.021306	1.000000	-0.084475	0.005205	0.053170	0.018685	-0.012642	0.012614	0.027990	0.055011	0.05898

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHO CONSUMIN
SMOKING	-0.036277	-0.084475	1.000000	-0.014585	0.160267	-0.042822	-0.141522	-0.029575	0.001913	-0.129426	-0.05062
YELLOW_FINGERS	0.212959	0.005205	-0.014585	1.000000	0.565829	0.323083	0.041122	-0.118058	-0.144300	-0.078515	-0.28902
ANXIETY	0.152127	0.053170	0.160267	0.565829	1.000000	0.216841	-0.009678	-0.188538	-0.165750	-0.191807	-0.16575
PEER_PRESSURE	0.275564	0.018685	-0.042822	0.323083	0.216841	1.000000	0.048515	0.078148	-0.081800	-0.068771	-0.15997
CHRONIC DISEASE	0.204606	-0.012642	-0.141522	0.041122	-0.009678	0.048515	1.000000	-0.110529	0.106386	-0.049967	0.00215
FATIGUE	0.083560	0.012614	-0.029575	-0.118058	-0.188538	0.078148	-0.110529	1.000000	0.003056	0.141937	-0.19137
ALLERGY	-0.154251	0.027990	0.001913	-0.144300	-0.165750	-0.081800	0.106386	0.003056	1.000000	0.173867	0.34433
WHEEZING	-0.141207	0.055011	-0.129426	-0.078515	-0.191807	-0.068771	-0.049967	0.141937	0.173867	1.000000	0.26565
ALCOHOL CONSUMING	-0.454268	0.058985	-0.050623	-0.289025	-0.165750	-0.159973	0.002150	-0.191377	0.344339	0.265659	1.00000
COUGHING	-0.133303	0.169950	-0.129471	-0.012640	-0.225644	-0.089019	-0.175287	0.146856	0.189524	0.374265	0.20272
SHORTNESS OF BREATH	0.064911	-0.017513	0.061264	-0.105944	-0.144077	-0.220175	-0.026459	0.441745	-0.030056	0.037834	-0.17941
SWALLOWING DIFFICULTY	0.078161	-0.001270	0.030718	0.345904	0.489403	0.366590	0.075176	-0.132790	-0.061508	0.069027	-0.00929
CHEST PAIN	-0.362958	-0.018104	0.120117	-0.104829	-0.113634	-0.094828	-0.036938	-0.010832	0.239433	0.147640	0.33122
LUNG_CANCER	0.067254	-0.089465	-0.058179	-0.181339	-0.144947	-0.186388	-0.110891	-0.150673	-0.327766	-0.249300	-0.28853

In [53]:
```
import seaborn as sns
sns.heatmap(data.corr(), annot=True)
```

Out[53]: <AxesSubplot:>



In [24]:
```
# Checking for null values
data.isnull().sum()
```

Out[24]:
```
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
```

Transformation

In [25]:
```
# label encoding
data.replace({"LUNG_CANCER":{'YES':0,'NO':1}},inplace=True)
# printing the first 5 rows of the dataframe
data.head(5)
```

Out[25]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTI OF BRI
0	M	69	1	2	2	1	1	2	1	2	2	2	
1	M	74	2	1	1	1	2	2	2	1	1	1	
2	F	59	1	1	1	2	1	2	1	2	1	2	
3	M	63	2	2	2	1	1	1	1	1	2	1	
4	F	63	1	2	1	1	1	1	1	2	1	2	

In [26]:
```
# Value_counts of loan_status
data['LUNG_CANCER'].value_counts()
```

Out[26]:
```
0    270
1     39
Name: LUNG_CANCER, dtype: int64
```

In [27]:
```
# label encoding
data.replace({"GENDER":{'M':0,'F':1}},inplace=True)
# printing the first 5 rows of the dataframe
data.head(5)
```

Out[27]:

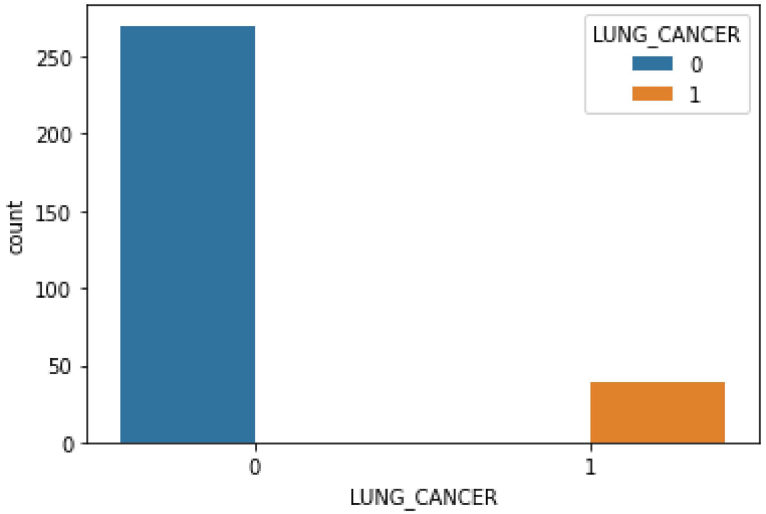
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTI OF BRI
0	0	69	1	2	2	1	1	2	1	2	2	2	
1	0	74	2	1	1	1	2	2	2	1	1	1	
2	1	59	1	1	1	2	1	2	1	2	1	2	
3	0	63	2	2	2	1	1	1	1	1	2	1	

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORT OF BRI
4	1	63	1	2	1	1	1	1	1	2	1	2	

Data Visualization

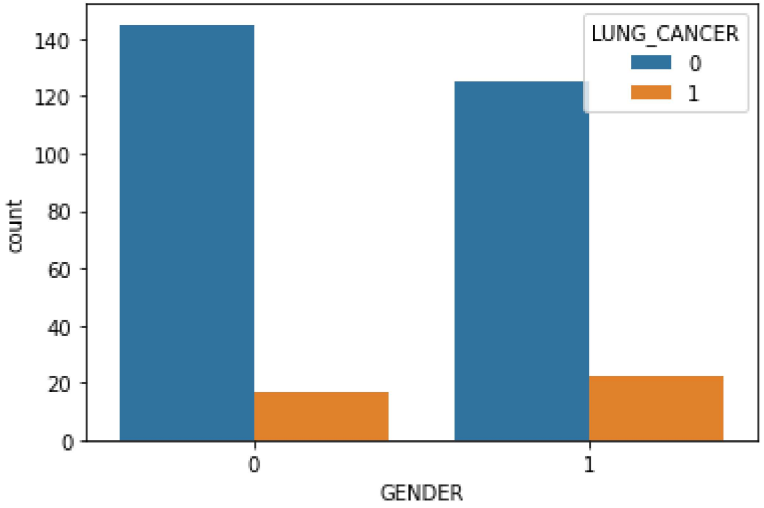
In [28]:
```
# education & Loan Status
import seaborn as sns
sns.countplot(x='LUNG_CANCER',hue='LUNG_CANCER',data=data)
```

Out[28]: <AxesSubplot:xlabel='LUNG_CANCER', ylabel='count'>



In [29]:
```
# education & Loan Status
import seaborn as sns
sns.countplot(x='GENDER',hue='LUNG_CANCER',data=data)
```

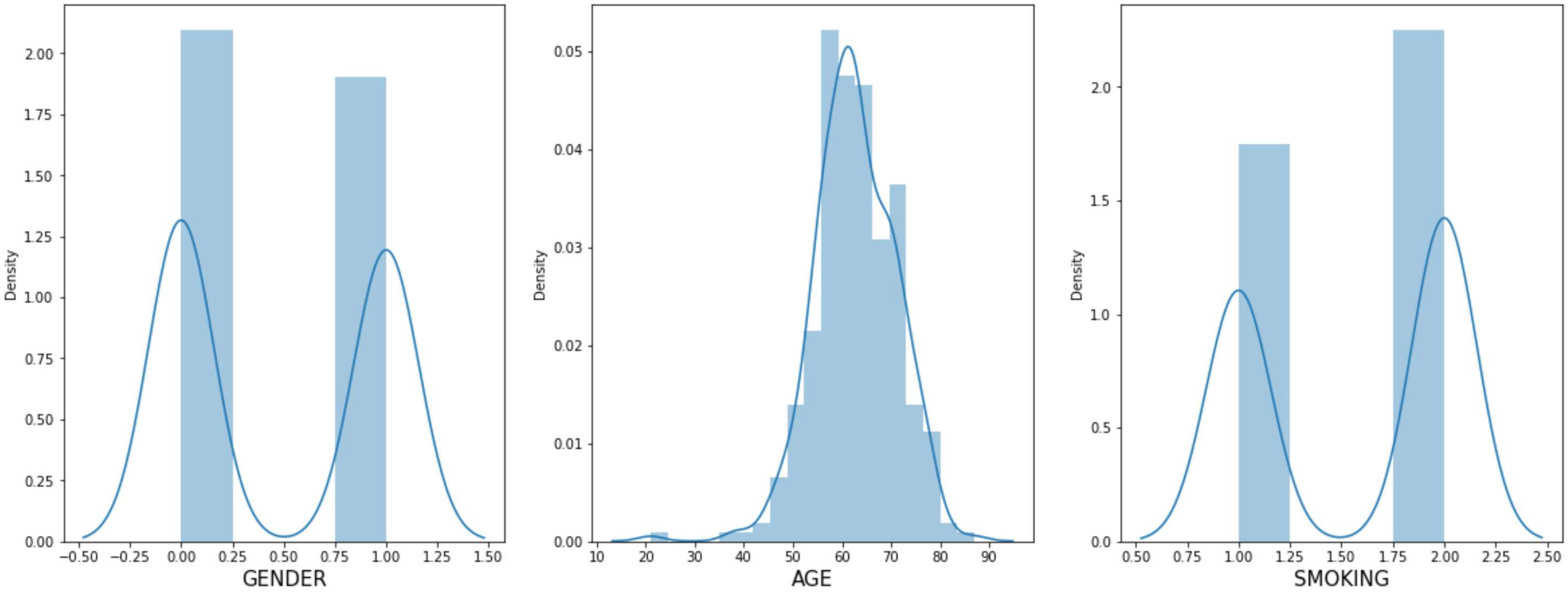
Out[29]: <AxesSubplot:xlabel='GENDER', ylabel='count'>

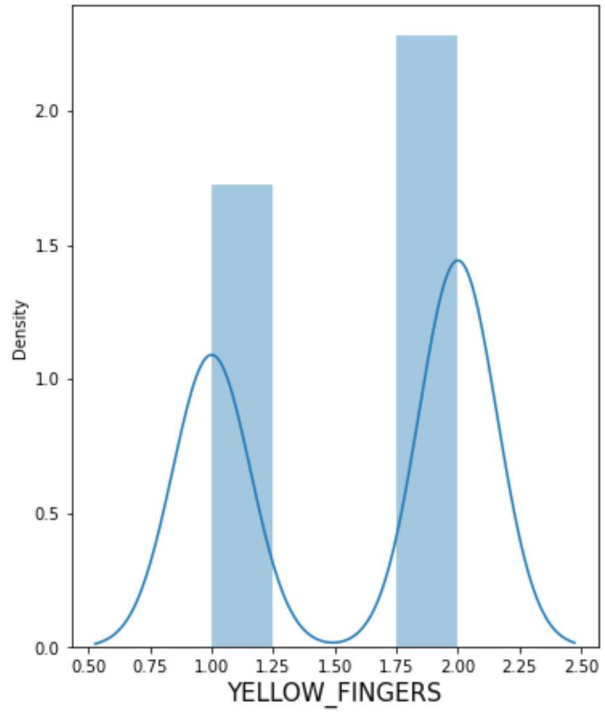


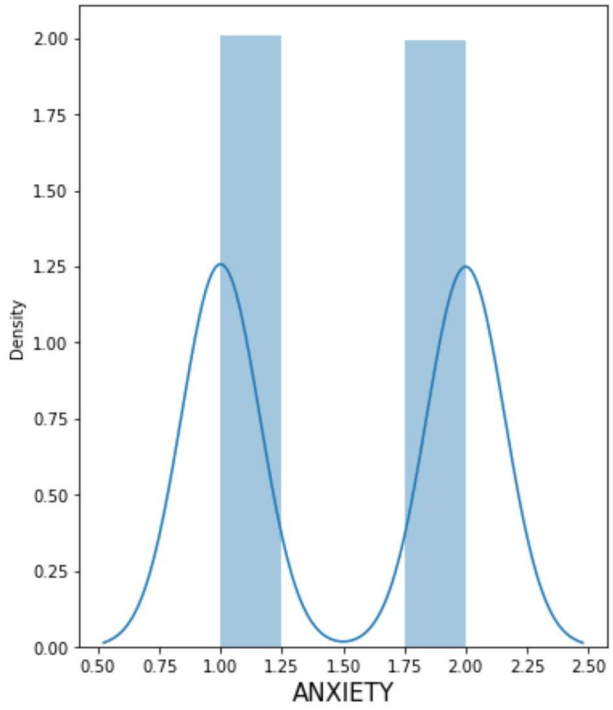
In [30]:
```
# let's see how data is distributed for every column
import matplotlib.pyplot as plt

plt.figure(figsize = (20, 25))
plotnumber = 1
for column in data:
    if plotnumber <= 9:
        ax = plt.subplot(3, 3, plotnumber)
        sns.distplot(data[column])
        plt.xlabel(column, fontsize = 15)

    plotnumber += 1
plt.show()
```

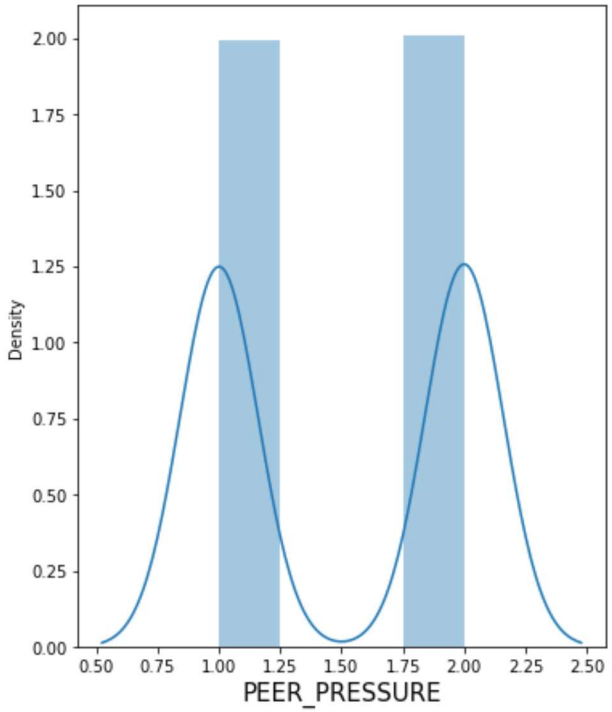
```
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
```

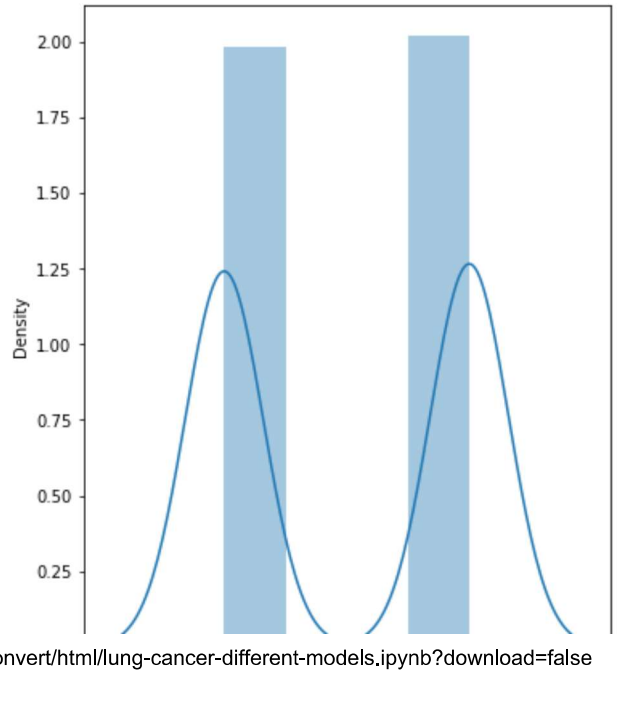
```
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Puneet Sharma\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

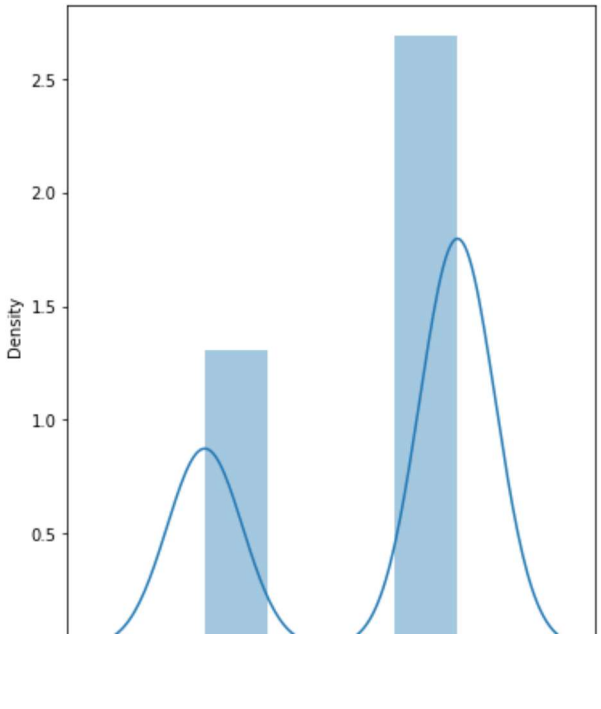


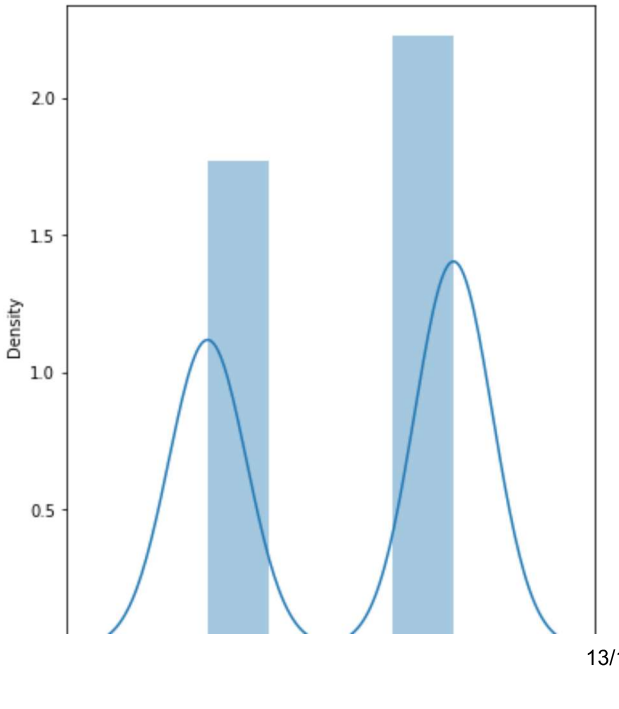

Density
YELLOW_FINGERS

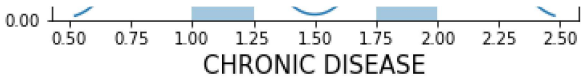

Density
ANXIETY

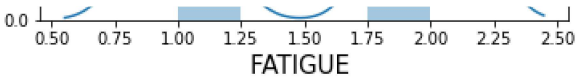

Density
PEER_PRESSURE

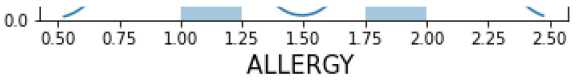

Density


Density


Density







We can see that the distribution of data is normal ! lets move for the Model preparation. 🚀

Model Preparation

Spilt into X & Y

In [31]:
```
# separating the data and target
X = data.drop(columns=['LUNG_CANCER'],axis=1)
y = data['LUNG_CANCER']
```

In [32]:
```
print("The shape of X is " , X.shape)
print("The shape of Y is " , y.shape)
```

```
The shape of X is  (309, 15)
The shape of Y is  (309,)
```

In [33]:
```
from sklearn.model_selection import train_test_split
# separating into train and testing
X_train, X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.2,stratify=y,random_state=42)
print("Shape of X_train is " ,X_train.shape)
print("Shape of X_test  is " ,X_test.shape)
print("Shape of Y_train is " ,Y_train.shape)
print("Shape of Y_test  is " ,Y_test.shape)
```

```
Shape of X_train is  (247, 15)
Shape of X_test  is  (62, 15)
Shape of Y_train is  (247,)
Shape of Y_test  is  (62,)
```

In [34]:
```
# After statify Y train & test values
print(Y_train.value_counts())
print(Y_test.value_counts())
```

```
0    216
1     31
Name: LUNG_CANCER, dtype: int64
0    54
1     8
Name: LUNG_CANCER, dtype: int64
```

Feature Scalilng

In [35]:
```
# scaling the data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [36]:
```
X_train
```

Out[36]:
```
array([[-0.89605787, -2.94594932,  0.9258201 , ...,  0.74394833,
        -0.95641449,  0.86704847],
       [-0.89605787, -0.34543211,  0.9258201 , ..., -1.34417937,
        -0.95641449,  0.86704847],
       [ 1.11599935,  0.1499045 , -1.08012345, ..., -1.34417937,
        -0.95641449, -1.15333806],
       ...,
       [-0.89605787, -0.34543211, -1.08012345, ..., -1.34417937,
        -0.95641449, -1.15333806],
       [ 1.11599935,  0.02607035, -1.08012345, ...,  0.74394833,
        -0.95641449, -1.15333806],
       [ 1.11599935, -0.46926626,  0.9258201 , ...,  0.74394833,
         1.04557178, -1.15333806]])
```

Model Training

We will train different model after the evaluation of model we will select out best model for production.

1. Logistic Regression
2. KNN
3. SVC
4. Decision Tree
5. Random Forest Regressor
6. XgBoost
7. Ada Boost
8. Gradient Boosting
9. Stochascated Gradient Boosting
10. Stacking

Logistic Regression

In [37]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

lr = LogisticRegression()
lr.fit(X_train, Y_train)
y_pred = lr.predict(X_test)

lr_train_acc = accuracy_score(Y_train, lr.predict(X_train))
lr_test_acc = accuracy_score(Y_test, y_pred)

print(f"Training Accuracy of Logistic Regression Model is {lr_train_acc}")
print(f"Test Accuracy of Logistic Regression Model is {lr_test_acc}")
```

```
Training Accuracy of Logistic Regression Model is 0.951417004048583
Test Accuracy of Logistic Regression Model is 0.9193548387096774
```

In [38]:
```
# confusion matrix
confusion_matrix(Y_test, y_pred)
```

Out[38]:
```
array([[53,  1],
       [ 4,  4]], dtype=int64)
```

In [39]:
```
# classification report
print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.93      0.98      0.95        54
           1       0.80      0.50      0.62         8

    accuracy                           0.92        62
   macro avg       0.86      0.74      0.79        62
weighted avg       0.91      0.92      0.91        62
```

In [ ]:
```

```