Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 07

Submitted by:

Name: MD Zuleyenine Ibne Noman

Roll No: 2002

Lab Date: 4 September,2023

Submission Date: 28 August,2023

Dataset related information:

- filename : survey lung cancer.csv
- Link of dataset : [https://www.kaggle.com/code/kelvinfoo123/predicting-lung-cancer-with-knn/input (https://www.kaggle.com/code/kelvinfoo123/predicting-lung-cancer-with-knn/input)](https://www.kaggle.com/code/kelvinfoo123/predicting-lung-cancer-with-knn/input)
- following are the Dataset characterizations:
    - Total no. of attributes:16
    - No .of instances:284 Attribute information: 1. Gender: M(male), F(female) 2. Age: Age of the patient 3. Smoking: YES=2 , NO=1. 4. Yellow fingers: YES=2 , NO=1. 5. Anxiety: YES=2 , NO=1. 6. Peer_pressure: YES=2 , NO=1. 7. Chronic Disease: YES=2 , NO=1. 8. Fatigue: YES=2 , NO=1. 9. Allergy: YES=2 , NO=1. 10. Wheezing: YES=2 , NO=1. 11. Alcohol: YES=2 , NO=1. 12. Coughing: YES=2 , NO=1. 13. Shortness of Breath: YES=2 , NO=1. 14. Swallowing Difficulty: YES=2 , NO=1. 15. Chest pain: YES=2 , NO=1. 16. Lung Cancer: YES , NO.

All needed installations:

Import Libraries

In [7]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Read the file /dataset

In [8]:
```
cancer=pd.read_csv("survey lung cancer.csv")
```

In [9]:
```
cancer
```
Out[9]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE
0	M	69	1	2	2	1	1
1	M	74	2	1	1	1	2
2	F	59	1	1	1	2	1
3	M	63	2	2	2	1	1
4	F	63	1	2	1	1	1
...	...	...	...	...	...	...	...
304	F	56	1	1	1	2	2
305	M	70	2	1	1	1	1
306	M	58	2	1	1	1	1
307	M	67	2	1	2	1	1
308	M	62	1	1	1	2	1

309 rows × 16 columns

Exploring the dataset

In [10]:
```
cancer.head()
```
Out[10]:

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

In [11]:
```
cancer.shape
```
Out[11]:
```
(309, 16)
```

In [12]:
```
cancer.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 [13]:
```
cancer.describe()
```

Out[13]:

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.00000
mean	62.673139	1.563107	1.569579	1.498382	1.501618	1.504854
std	8.210301	0.496806	0.495938	0.500808	0.500808	0.500787
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.00000
25%	57.000000	1.000000	1.000000	1.000000	1.000000	1.00000
50%	62.000000	2.000000	2.000000	1.000000	2.000000	2.00000
75%	69.000000	2.000000	2.000000	2.000000	2.000000	2.00000
max	87.000000	2.000000	2.000000	2.000000	2.000000	2.00000

Dataset preprocessing

!pip install sklearn !pip install scikit-learn

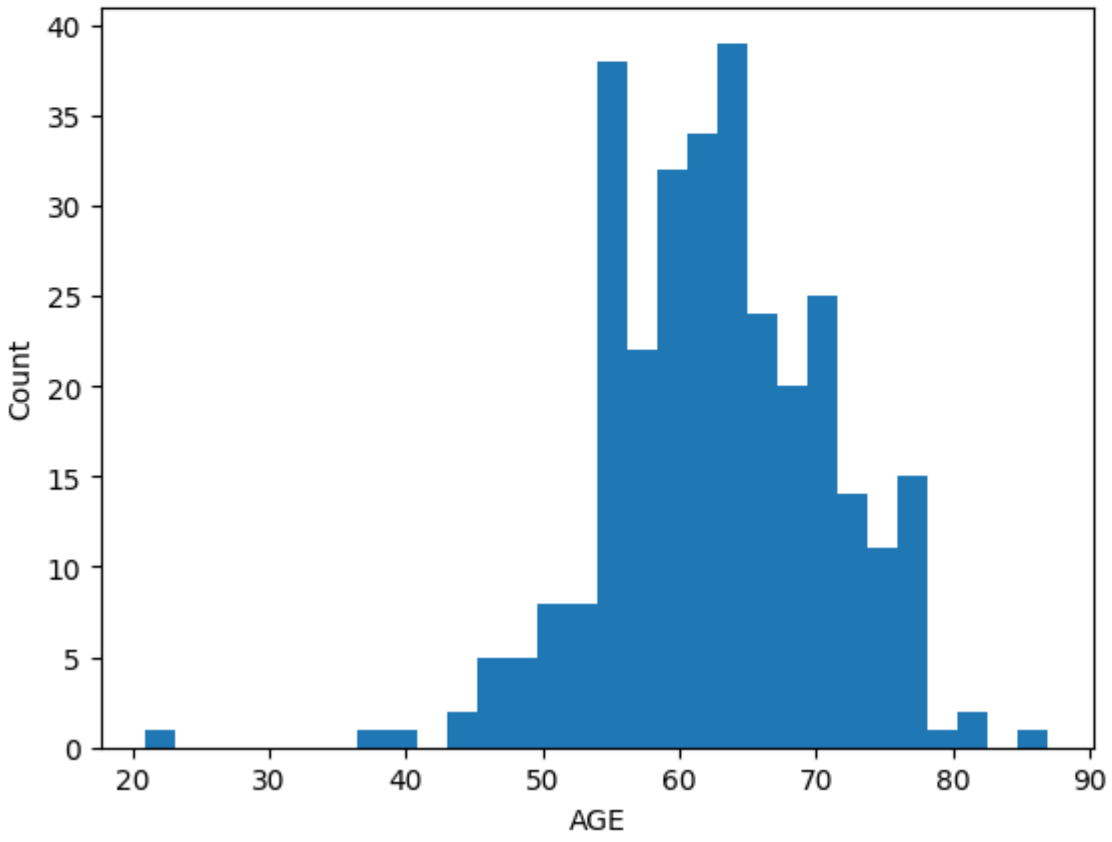
In [14]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
cancer["LUNG_CANCER"]=lb.fit_transform(cancer["LUNG_CANCER"])
```

In [15]:
```
cancer["GENDER"]=lb.fit_transform(cancer["GENDER"])
```

In [16]:
```
cancer["LUNG_CANCER"].value_counts()
```

Out[16]:
```
LUNG_CANCER
1    270
0     39
Name: count, dtype: int64
```

In [17]:
```
plt.hist(cancer["AGE"],bins=30)
plt.xlabel("AGE")
plt.ylabel("Count")
plt.show()
```



In [18]:
```
cancer=cancer[cancer.AGE>30]
```

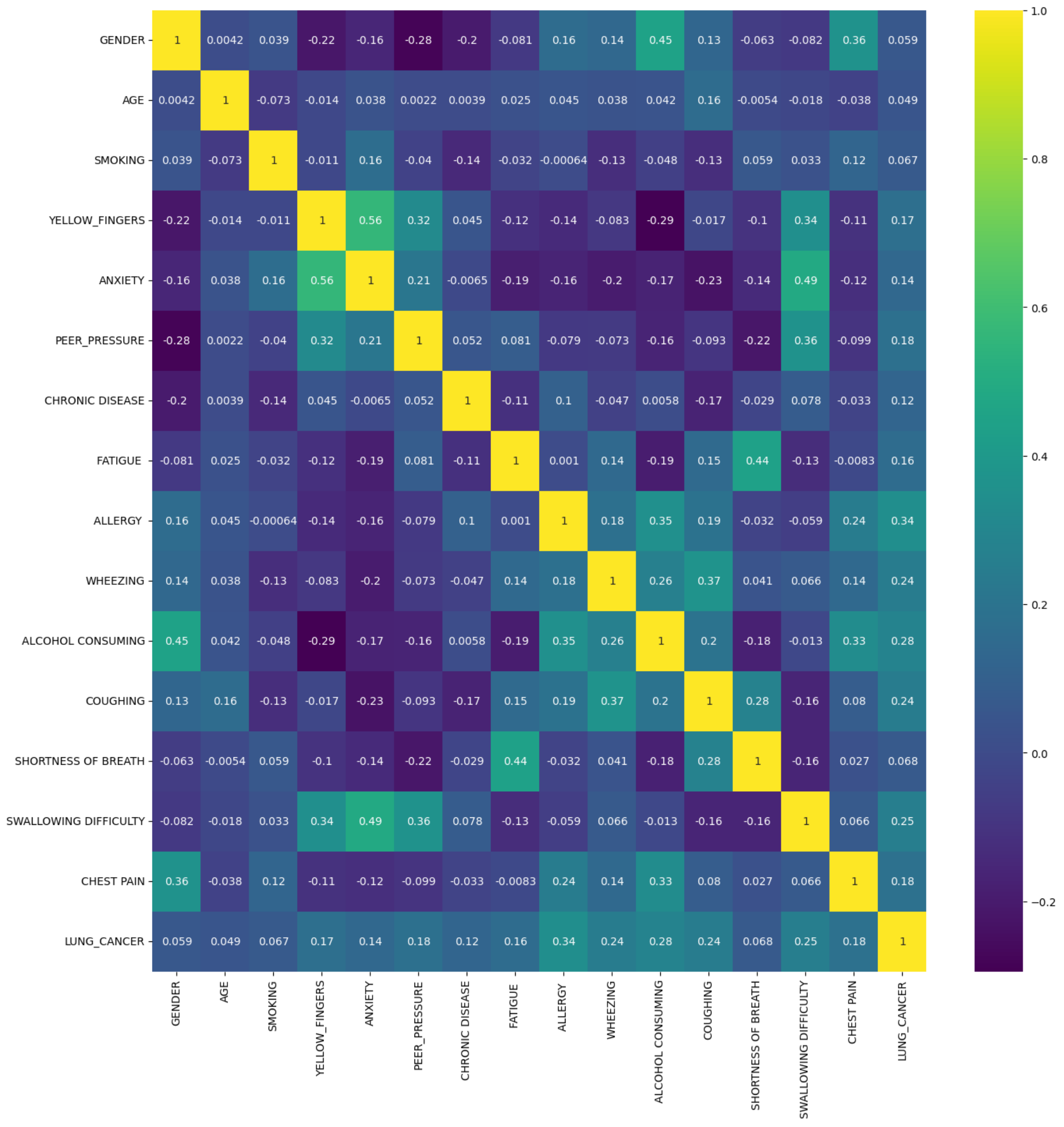
In [19]:
```
cancer_without_age = cancer.drop(["AGE"], axis = 1)
for i in cancer_without_age.columns:
  print(cancer_without_age[i].value_counts())
```

```
GENDER
1    162
0    146
Name: count, dtype: int64
SMOKING
2    173
1    135
Name: count, dtype: int64
YELLOW_FINGERS
2    176
1    132
Name: count, dtype: int64
ANXIETY
2    154
1    154
Name: count, dtype: int64
PEER_PRESSURE
2    155
1    153
Name: count, dtype: int64
CHRONIC DISEASE
2    155
1    153
Name: count, dtype: int64
FATIGUE
2    207
1    101
Name: count, dtype: int64
ALLERGY
2    171
1    137
Name: count, dtype: int64
WHEEZING
2    172
1    136
Name: count, dtype: int64
ALCOHOL CONSUMING
2    172
1    136
Name: count, dtype: int64
COUGHING
2    179
1    129
Name: count, dtype: int64
SHORTNESS OF BREATH
2    197
1    111
Name: count, dtype: int64
SWALLOWING DIFFICULTY
1    163
2    145
Name: count, dtype: int64
CHEST PAIN
2    172
```

```
1   136
Name: count, dtype: int64
LUNG_CANCER
1   270
0    38
Name: count, dtype: int64
```

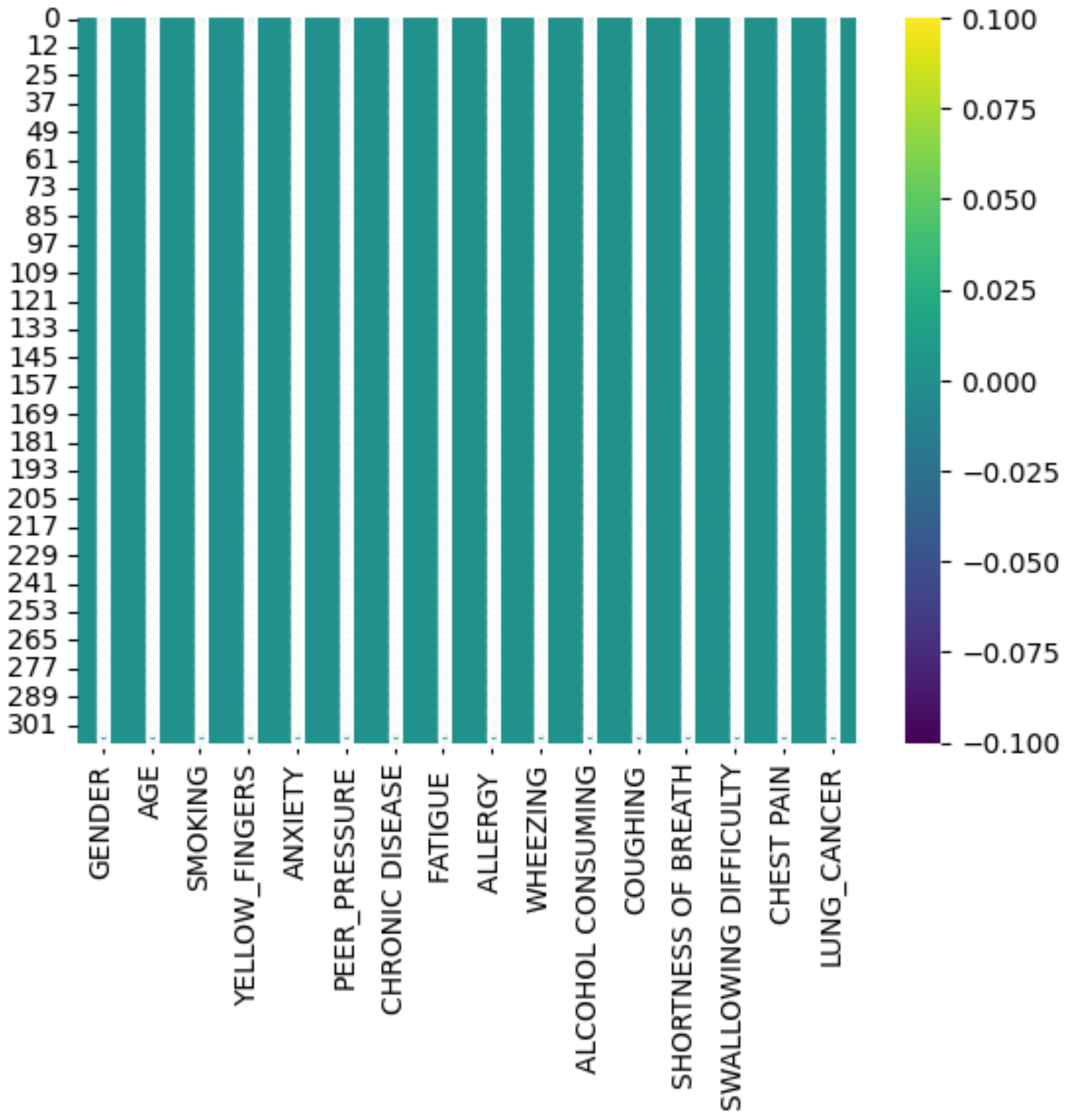
In [20]:
```
plt.figure(figsize=(16,16))
sns.heatmap(cancer.corr(),annot=True,cmap="viridis")
```

Out[20]:
```
<AxesSubplot: >
```



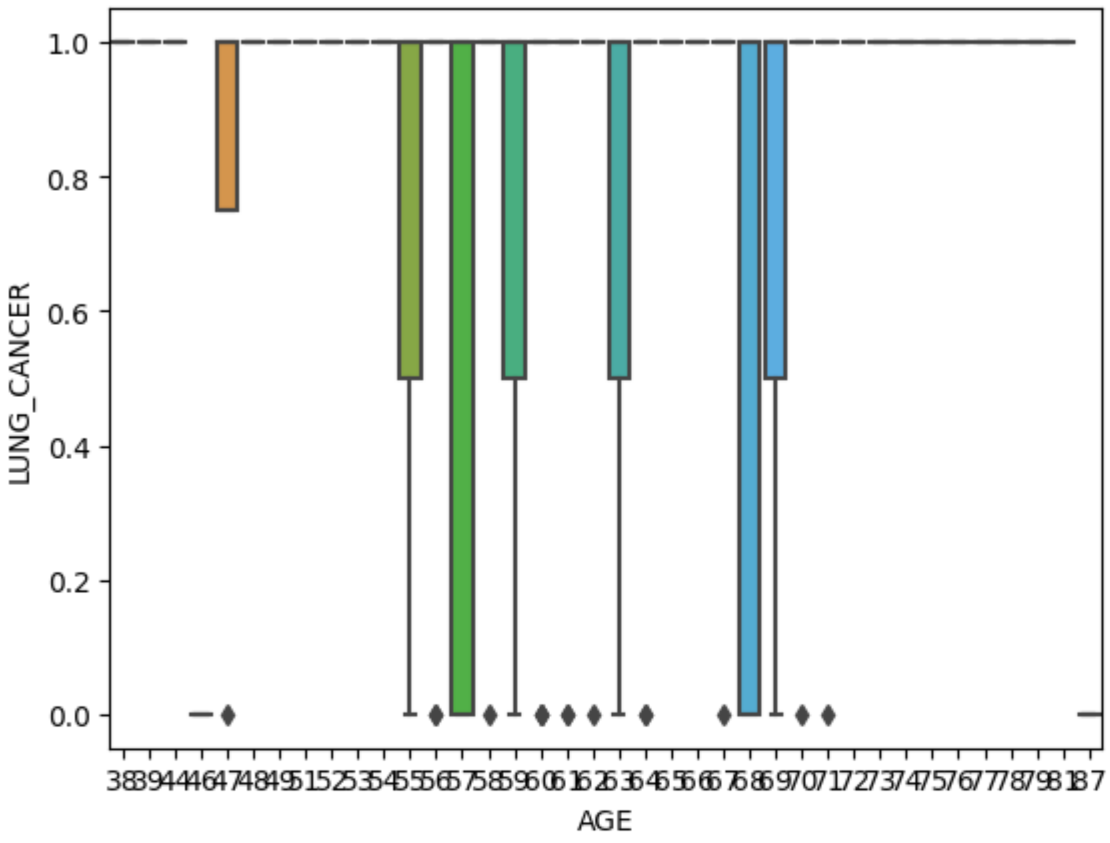
In [21]:
```
sns.heatmap(cancer.isnull(),annot=True,cmap="viridis")
```

Out[21]:
```
<AxesSubplot: >
```



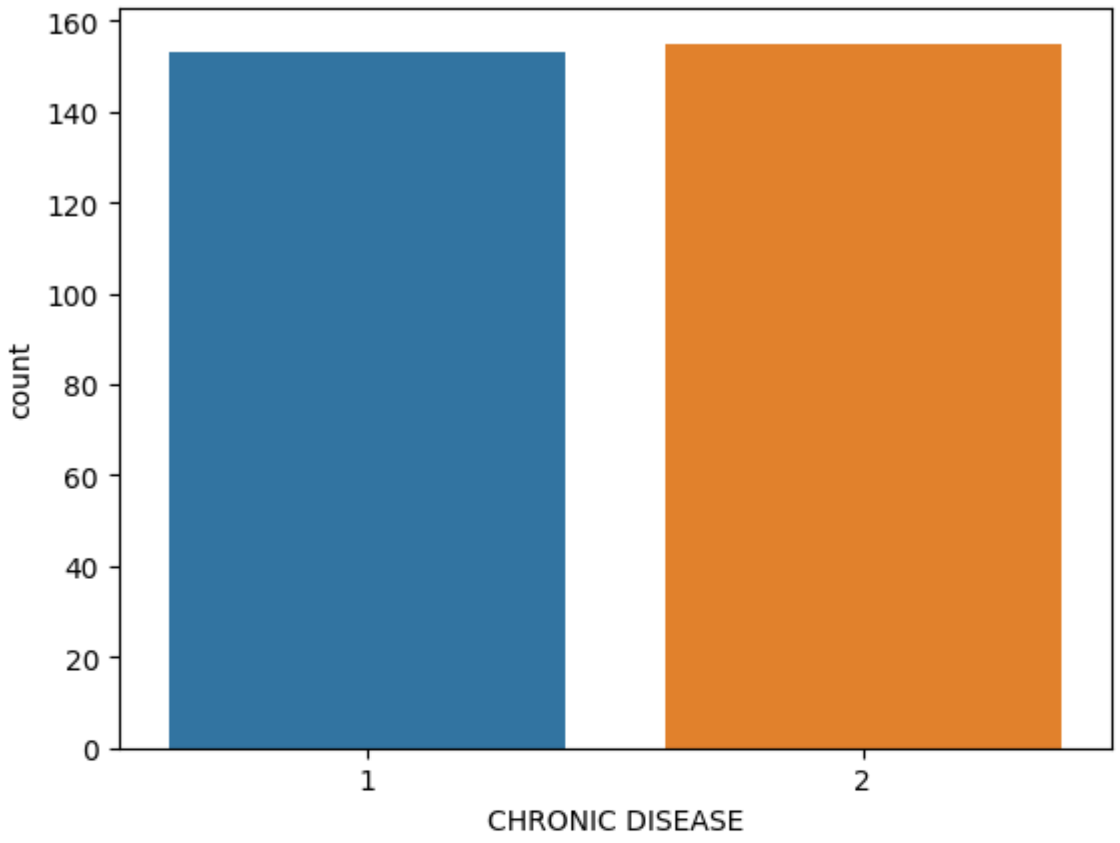
```
In [22]: sns.boxplot(x="AGE",y="LUNG_CANCER",data=cancer)
```

```
Out[22]: <AxesSubplot: xlabel='AGE', ylabel='LUNG_CANCER'>
```



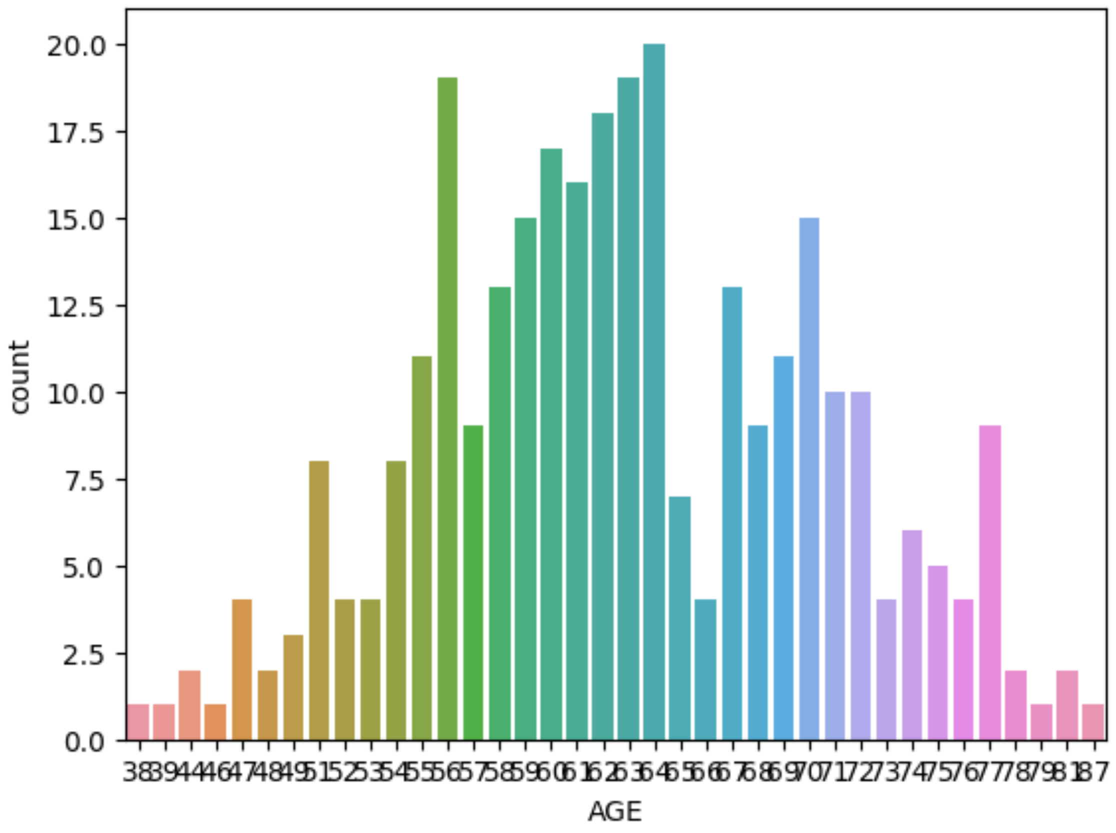
In [23]:
```
sns.countplot(x="CHRONIC DISEASE",data=cancer)
```

Out[23]:
```
<AxesSubplot: xlabel='CHRONIC DISEASE', ylabel='count'>
```



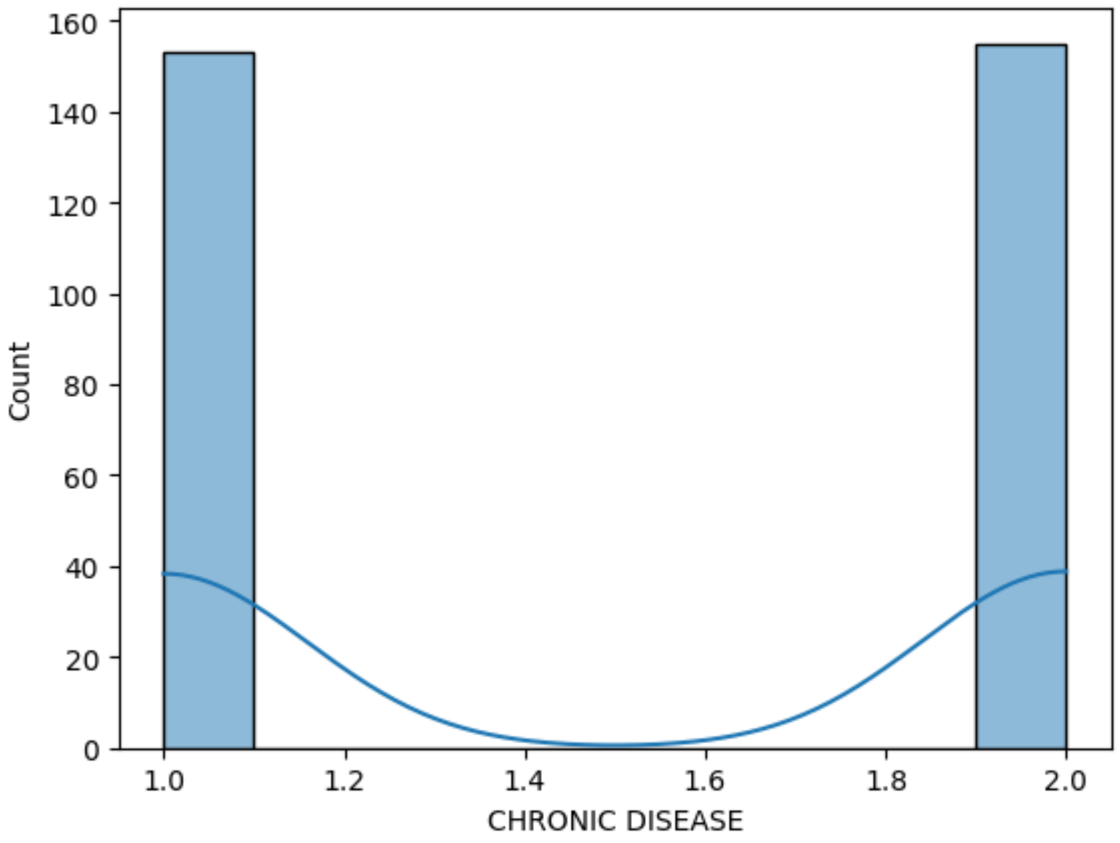
In [24]:
```
sns.countplot(x="AGE",data=cancer)
```

Out[24]:
```
<AxesSubplot: xlabel='AGE', ylabel='count'>
```



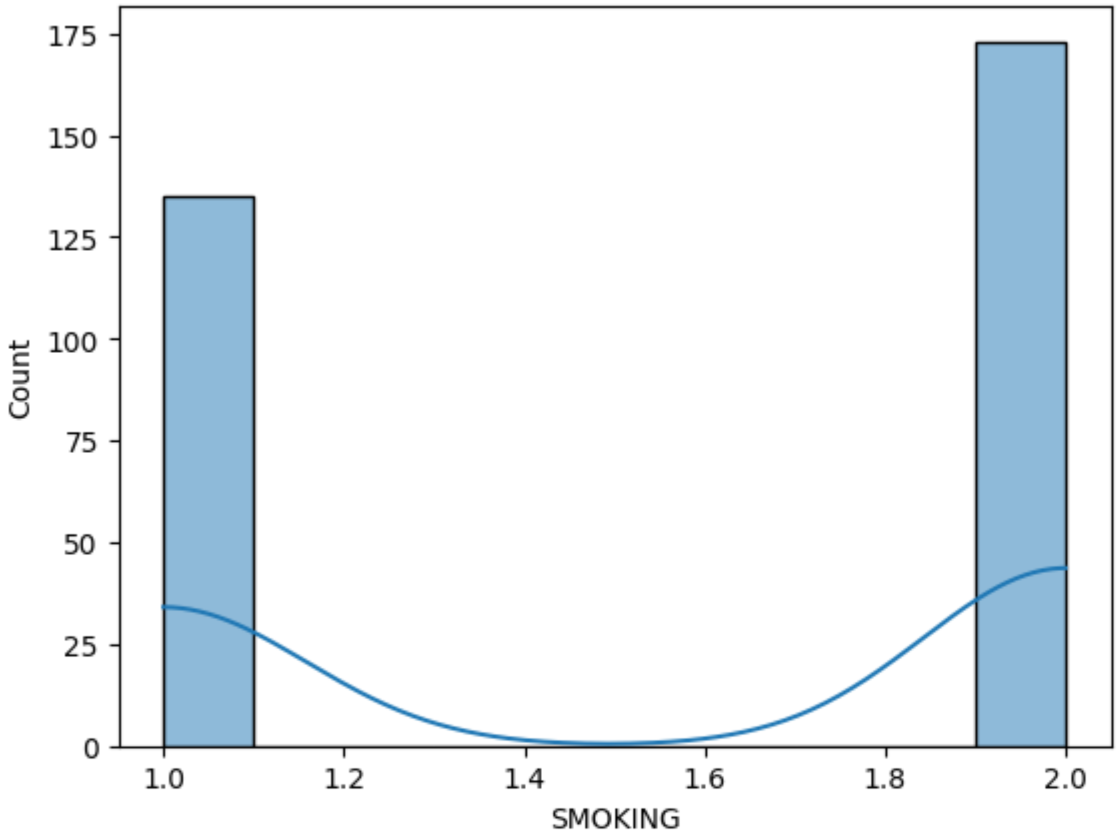
```
In [25]: sns.histplot(cancer["CHRONIC DISEASE"],bins=10,kde=True)
```

```
Out[25]: <AxesSubplot: xlabel='CHRONIC DISEASE', ylabel='Count'>
```



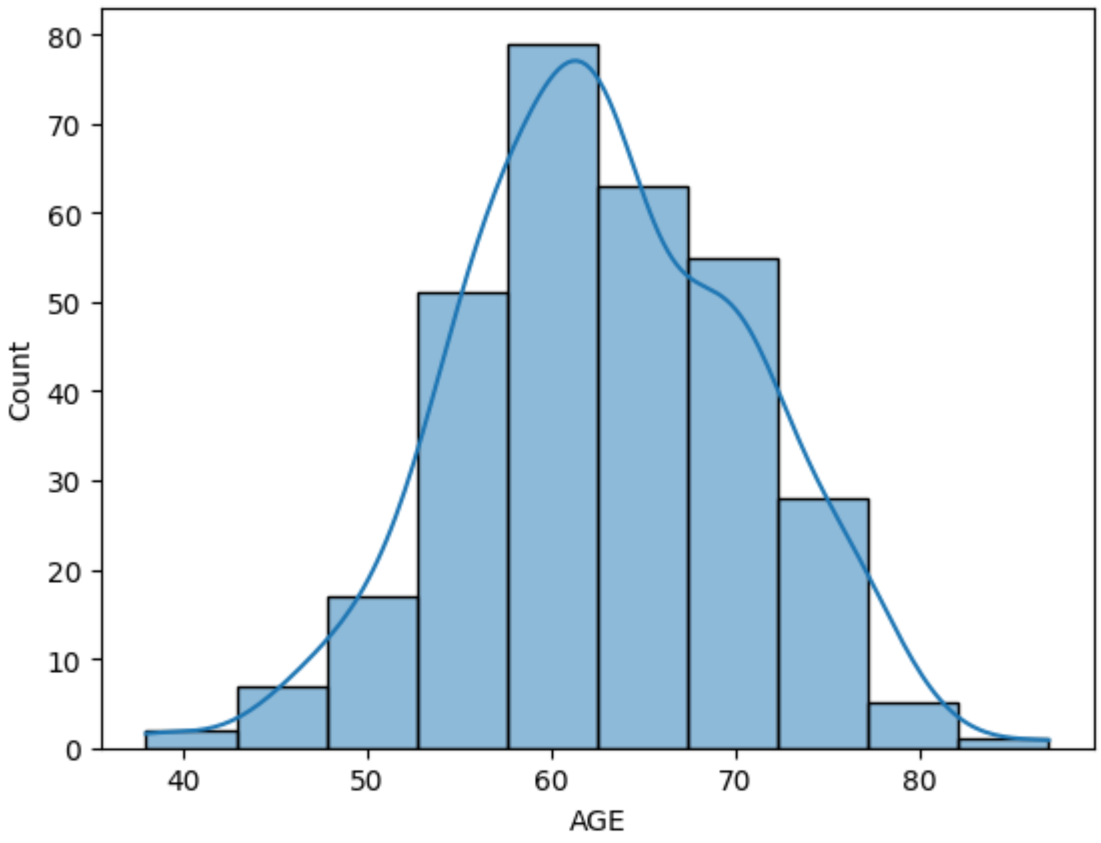
```
In [26]: sns.histplot(cancer["SMOKING"],bins=10,kde=True)
```

```
Out[26]: <AxesSubplot: xlabel='SMOKING', ylabel='Count'>
```

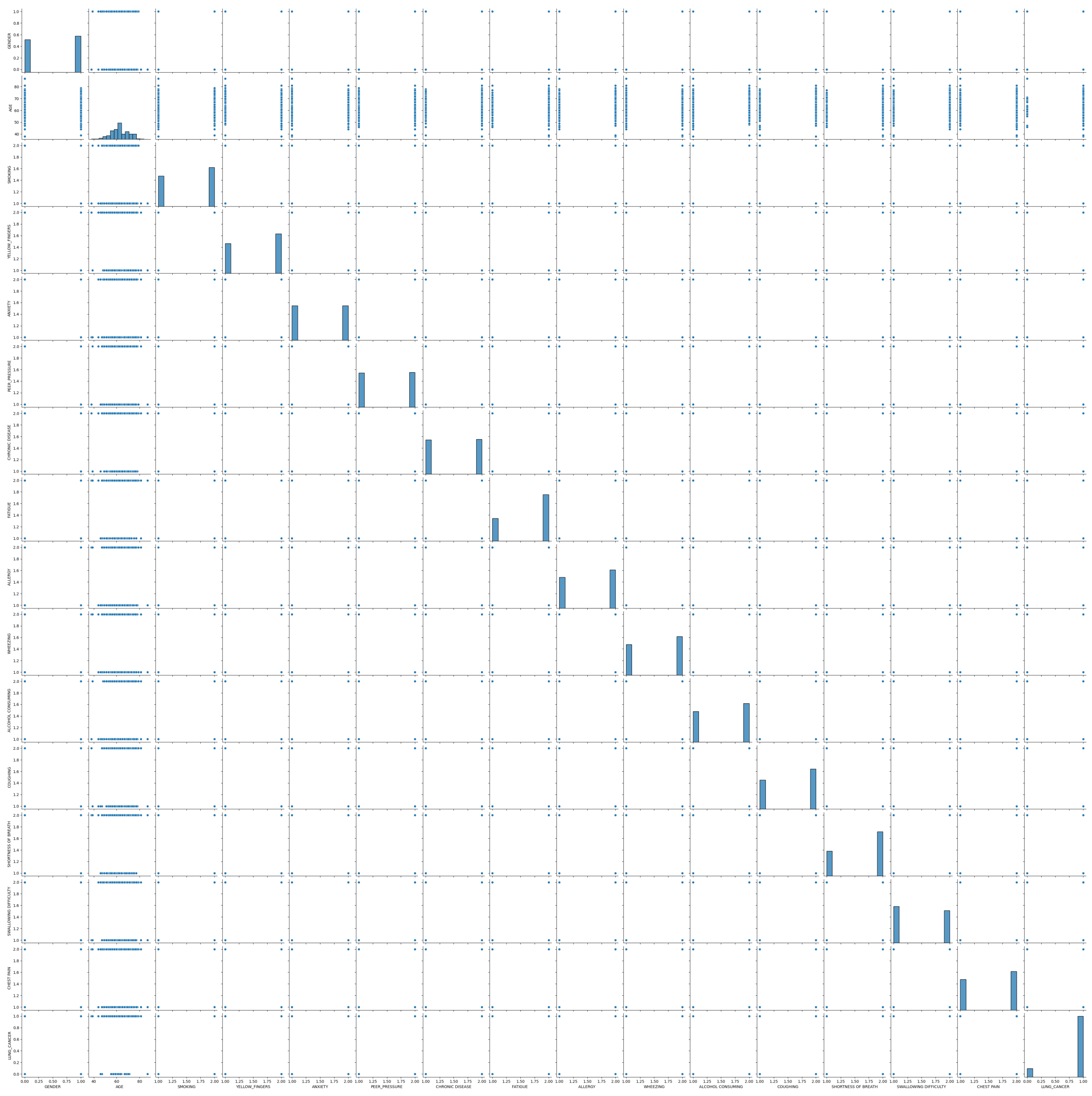


In [27]:
```
sns.histplot(cancer["AGE"],bins=10,kde=True)
```

Out[27]:
```
<AxesSubplot: xlabel='AGE', ylabel='Count'>
```



In [28]:
```
sns.pairplot(cancer)
plt.show()
```



SVM model Training and Evaluation

In [29]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_sco
from sklearn.metrics import confusion_matrix, classification_report
```

In [30]:
```
x_train, x_test, y_train, y_test = train_test_split(cancer.drop('LUNG_
```

In [31]:
```
from sklearn.svm import SVC
model=SVC()
model.fit(x_train,y_train)
```

Out[31]:
```
▾ SVC
SVC()
```

In [32]:
```
predictions=model.predict(x_test)
```

In [33]:
```
print(confusion_matrix(y_test,predictions))
```

```
[[ 0 11]
 [ 0 82]]
```

In [34]:
```
print(accuracy_score(y_test,predictions))
```

```
0.8817204301075269
```

In [35]:
```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        11
           1       0.88      1.00      0.94        82

    accuracy                           0.88        93
   macro avg       0.44      0.50      0.47        93
weighted avg       0.78      0.88      0.83        93


/home/eyenine/.local/lib/python3.10/site-packages/sklearn/metrics/
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted samples. U
se `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/eyenine/.local/lib/python3.10/site-packages/sklearn/metrics/
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted samples. U
se `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/home/eyenine/.local/lib/python3.10/site-packages/sklearn/metrics/
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted samples. U
se `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [36]:
```
print(precision_score(y_test, predictions))
```

```
0.8817204301075269
```

In [37]:
```
print(recall_score(y_test, predictions))
```

```
1.0
```

In [38]:
```
print(f1_score(y_test, predictions))
```

```
0.9371428571428571
```

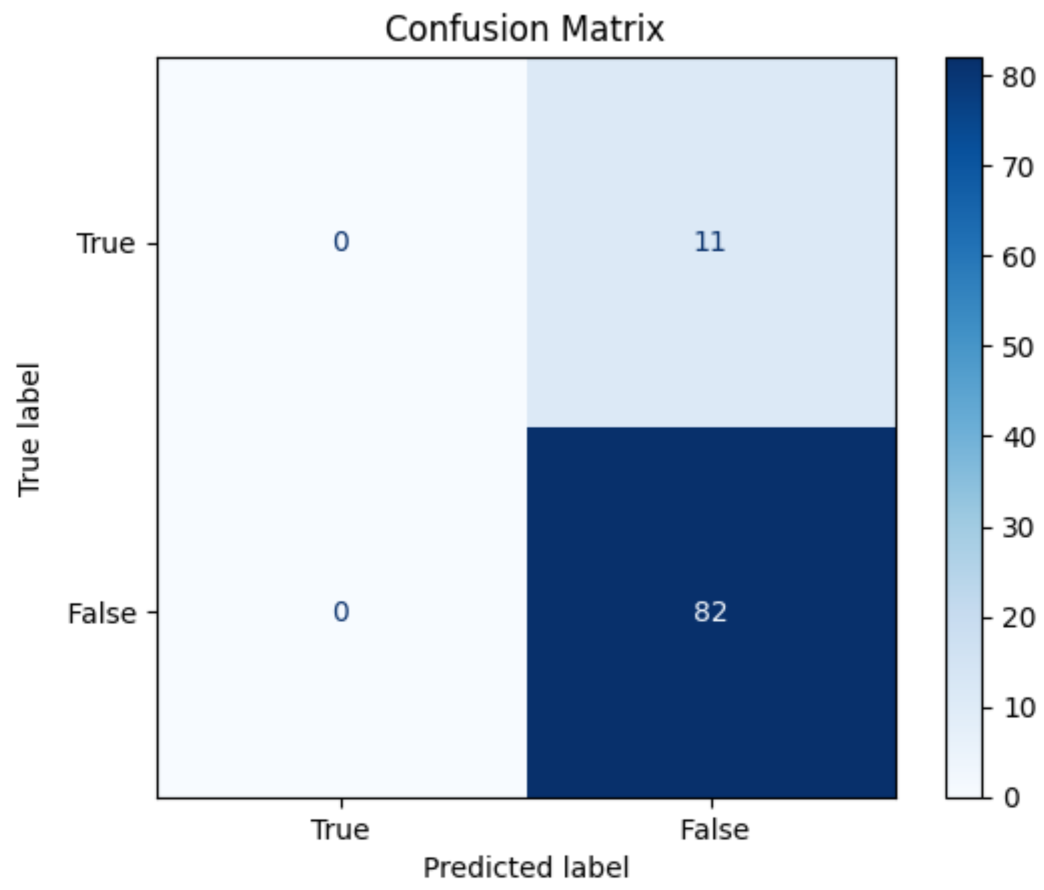
In [39]:
```
print(confusion_matrix(y_test, predictions))
```

```
[[ 0 11]
 [ 0 82]]
```

In [40]:
```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay


cm = confusion_matrix(y_test, predictions)


disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["T
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.show()
```


Confusion Matrix
True
False
True label
True
False
Predicted label