lung-cancer-code

May 25, 2025

[1]:

```python
import pandas as pd
import matplotlib. pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

[2]:

```python
lung_data = pd.read_csv("c:/Users/shrut/Downloads/survey lung cancer.csv")
```

[3]:

```python
lung_data
```

[3]:

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

     CHRONIC DISEASE  FATIGUE  ALLERGY  WHEEZING  ALCOHOL CONSUMING  COUGHING   \
0                  1        2        1         2                  2         2
1                  2        2        2         1                  1         1
2                  1        2        1         2                  1         2
3                  1        1        1         1                  2         1
4                  1        1        1         2                  1         2
..               …      …      …       …                …       …
304                2        2        1         1                  2         2
305                1        2        2         2                  2         2
306                1        1        2         2                  2         2
307                1        2        2         1                  2         2
308                1        2        2         2                  2         1

     SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN LUNG_CANCER
```

```
0                       2                      2           2         YES
1                       2                      2           2         YES
2                       2                      1           2          NO
3                       1                      2           2          NO
4                       2                      1           1          NO
..                   …                    …         …         …
304                     2                      2           1         YES
305                     2                      1           2         YES
306                     1                      1           2         YES
307                     2                      1           2         YES
308                     1                      2           1         YES

[309 rows x 16 columns]
```

```
[4]: lung_data.head()
```

```
[4]:   GENDER  AGE  SMOKING  YELLOW_FINGERS  ANXIETY  PEER_PRESSURE  \
     0      M   69        1               2        2              1
     1      M   74        2               1        1              1
     2      F   59        1               1        1              2
     3      M   63        2               2        2              1
     4      F   63        1               2        1              1

        CHRONIC DISEASE  FATIGUE  ALLERGY  WHEEZING  ALCOHOL CONSUMING  COUGHING  \
     0                1        2        1         2                  2         2
     1                2        2        2         1                  1         1
     2                1        2        1         2                  1         2
     3                1        1        1         1                  2         1
     4                1        1        1         2                  1         2

        SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN LUNG_CANCER
     0                    2                      2           2         YES
     1                    2                      2           2         YES
     2                    2                      1           2          NO
     3                    1                      2           2          NO
     4                    2                      1           1          NO
```

```
[5]: lung_data.tail()
```

```
[5]:     GENDER  AGE  SMOKING  YELLOW_FINGERS  ANXIETY  PEER_PRESSURE  \
     304      F   56        1               1        1              2
     305      M   70        2               1        1              1
     306      M   58        2               1        1              1
     307      M   67        2               1        2              1
     308      M   62        1               1        1              2

          CHRONIC DISEASE  FATIGUE  ALLERGY  WHEEZING  ALCOHOL CONSUMING  COUGHING  \
304                2        2        1         1                  2         2
305                1        2        2         2                  2         2
306                1        1        2         2                  2         2
307                1        2        2         1                  2         2
308                1        2        2         2                  2         1

     SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN LUNG_CANCER
304                    2                      2           1         YES
305                    2                      1           2         YES
306                    1                      1           2         YES
307                    2                      1           2         YES
308                    1                      2           1         YES
```

[6]:

```
#dependent_variable
x = lung_data.iloc[:,0:-1]
print(x)
```

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

     CHRONIC DISEASE  FATIGUE  ALLERGY  WHEEZING  ALCOHOL CONSUMING  COUGHING  \
0                  1        2        1         2                  2         2
1                  2        2        2         1                  1         1
2                  1        2        1         2                  1         2
3                  1        1        1         1                  2         1
4                  1        1        1         2                  1         2
..               ...      ...      ...       ...                ...       ...
304                2        2        1         1                  2         2
305                1        2        2         2                  2         2
306                1        1        2         2                  2         2
307                1        2        2         1                  2         2
308                1        2        2         2                  2         1

     SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN
0                      2                      2           2
1                      2                      2           2
2                      2                      1           2
```

```
3                        1                      2            2
4                        2                      1            1
..                     …                    …          …
304                      2                      2            1
305                      2                      1            2
306                      1                      1            2
307                      2                      1            2
308                      1                      2            1

[309 rows x 15 columns]
```

[7]:
```
#independent_variable
y = lung_data. iloc[:,-1:]
print(y)
```

```
    LUNG_CANCER
0            YES
1            YES
2             NO
3             NO
4             NO
..           …
304          YES
305          YES
306          YES
307          YES
308          YES

[309 rows x 1 columns]
```

[8]:
```
lung_data.GENDER = lung_data.GENDER.map({"M":1,"F":2})
lung_data.LUNG_CANCER = lung_data.LUNG_CANCER.map({"YES":1,"NO":2})
```

[9]:
```
lung_data.shape
```

[9]:
```
(309, 16)
```

[10]:
```
lung_data.isnull().sum()
```

[10]:
```
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
```

```
[11]: lung_data.dtypes
```

```
[11]: 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
```

```
[12]: #the describe() method returns description of data in DataFrame
      lung_data.describe()
```

```
[12]:            GENDER         AGE     SMOKING  YELLOW_FINGERS     ANXIETY  \
      count  309.000000  309.000000  309.000000      309.000000  309.000000
      mean     1.475728   62.673139    1.563107        1.569579    1.498382
      std      0.500221    8.210301    0.496806        0.495938    0.500808
      min      1.000000   21.000000    1.000000        1.000000    1.000000
      25%      1.000000   57.000000    1.000000        1.000000    1.000000
      50%      1.000000   62.000000    2.000000        2.000000    1.000000
      75%      2.000000   69.000000    2.000000        2.000000    2.000000
      max      2.000000   87.000000    2.000000        2.000000    2.000000

             PEER_PRESSURE  CHRONIC DISEASE     FATIGUE     ALLERGY    WHEEZING  \
      count     309.000000       309.000000  309.000000  309.000000  309.000000
      mean        1.501618         1.504854    1.673139    1.556634    1.556634
      std         0.500808         0.500787    0.469827    0.497588    0.497588
```

```
min         1.000000         1.000000    1.000000    1.000000    1.000000
25%         1.000000         1.000000    1.000000    1.000000    1.000000
50%         2.000000         2.000000    2.000000    2.000000    2.000000
75%         2.000000         2.000000    2.000000    2.000000    2.000000
max         2.000000         2.000000    2.000000    2.000000    2.000000

       ALCOHOL CONSUMING    COUGHING  SHORTNESS OF BREATH  \
count         309.000000  309.000000           309.000000
mean            1.556634    1.579288             1.640777
std             0.497588    0.494474             0.480551
min             1.000000    1.000000             1.000000
25%             1.000000    1.000000             1.000000
50%             2.000000    2.000000             2.000000
75%             2.000000    2.000000             2.000000
max             2.000000    2.000000             2.000000

       SWALLOWING DIFFICULTY  CHEST PAIN  LUNG_CANCER
count             309.000000  309.000000   309.000000
mean                1.469256    1.556634     1.126214
std                 0.499863    0.497588     0.332629
min                 1.000000    1.000000     1.000000
25%                 1.000000    1.000000     1.000000
50%                 1.000000    2.000000     1.000000
75%                 2.000000    2.000000     1.000000
max                 2.000000    2.000000     2.000000
```

[13]:

```
#the info() method prints information of the database
lung_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    int64
 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    int64
dtypes: int64(16)
memory usage: 38.8 KB
```

[14]:
```
#Splitting the Dataset: Training and Testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/
 ↪3,random_state=0)
```

[15]:
```
lung_data['LUNG_CANCER'].value_counts()
```

[15]:
```
LUNG_CANCER
1    270
2     39
Name: count, dtype: int64
```

[16]:
```
len(lung_data)
```

[16]:
```
309
```

[17]:
```
#dependent_variable
x = lung_data.iloc[:,0:-1]
x
```

[17]:
```
     GENDER  AGE  SMOKING  YELLOW_FINGERS  ANXIETY  PEER_PRESSURE  \
0         1   69        1               2        2              1
1         1   74        2               1        1              1
2         2   59        1               1        1              2
3         1   63        2               2        2              1
4         2   63        1               2        1              1
..      …    …      …             …      …            …
304       2   56        1               1        1              2
305       1   70        2               1        1              1
306       1   58        2               1        1              1
307       1   67        2               1        2              1
308       1   62        1               1        1              2

     CHRONIC DISEASE  FATIGUE   ALLERGY   WHEEZING  ALCOHOL CONSUMING  COUGHING  \
0                  1         2         1          2                  2         2
1                  2         2         2          1                  1         1
2                  1         2         1          2                  1         2
3                  1         1         1          1                  2         1
4                  1         1         1          2                  1         2
..               …       …       …        …                …       …
304                2         2         1          1                  2         2
```

```
305                1        2        2         2                   2         2
306                1        1        2         2                   2         2
307                1        2        2         1                   2         2
308                1        2        2         2                   2         1

     SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN
0                      2                      2           2
1                      2                      2           2
2                      2                      1           2
3                      1                      2           2
4                      2                      1           1
..                   …                    …         …
304                    2                      2           1
305                    2                      1           2
306                    1                      1           2
307                    2                      1           2
308                    1                      2           1

[309 rows x 15 columns]
```

[18]:

```
#independent_variable
y = lung_data.iloc[:,-1:]
y
```

[18]:

```
     LUNG_CANCER
0              1
1              1
2              2
3              2
4              2
..           …
304            1
305            1
306            1
307            1
308            1

[309 rows x 1 columns]
```

[19]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

[20]:

```
from sklearn.linear_model import LogisticRegression
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/
 ↪3,random_state=0)
```

[21]:

```
#Fitting simple linear regression to the training test
Model1 = LogisticRegression()
Model1.fit(x_train, y_train)
#Predicting the test set results
prediction1 = Model1.predict(x_test)
```

```
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\utils\validation.py:1408: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\linear_model\_logistic.py:465: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[22]:

```
prediction1
```

[22]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1,
       1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

[23]:

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
confusion_matrix(y_test,prediction1)
```

[23]:

```
array([[85,  2],
       [10,  6]], dtype=int64)
```

[24]:

```
accuracy_score(y_test,prediction1)
```

[24]:

```
0.883495145631068
```

[25]:

```
from sklearn.metrics import precision_score
probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction1, average = None)
```

[25]:

```
array([0.89473684, 0.75      ])
```

[26]:

```
from sklearn.metrics import precision_score, recall_score, f1_score

# assuming your predicted and actual labels are stored in variables y_pred and␣
 ↪y_true, respectively
accuracy = accuracy_score(y_test, prediction1)
precision = precision_score(y_test, prediction1)
recall = recall_score(y_test, prediction1)
f1 = f1_score(y_test, prediction1)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.883495145631068
Precision: 0.8947368421052632
Recall: 0.9770114942528736
F1 score: 0.9340659340659341
```

[27]:

```
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

[28]:

```
recall_score(y_test, prediction1, average = None)
```

[28]:

```
array([0.97701149, 0.375     ])
```

[29]:

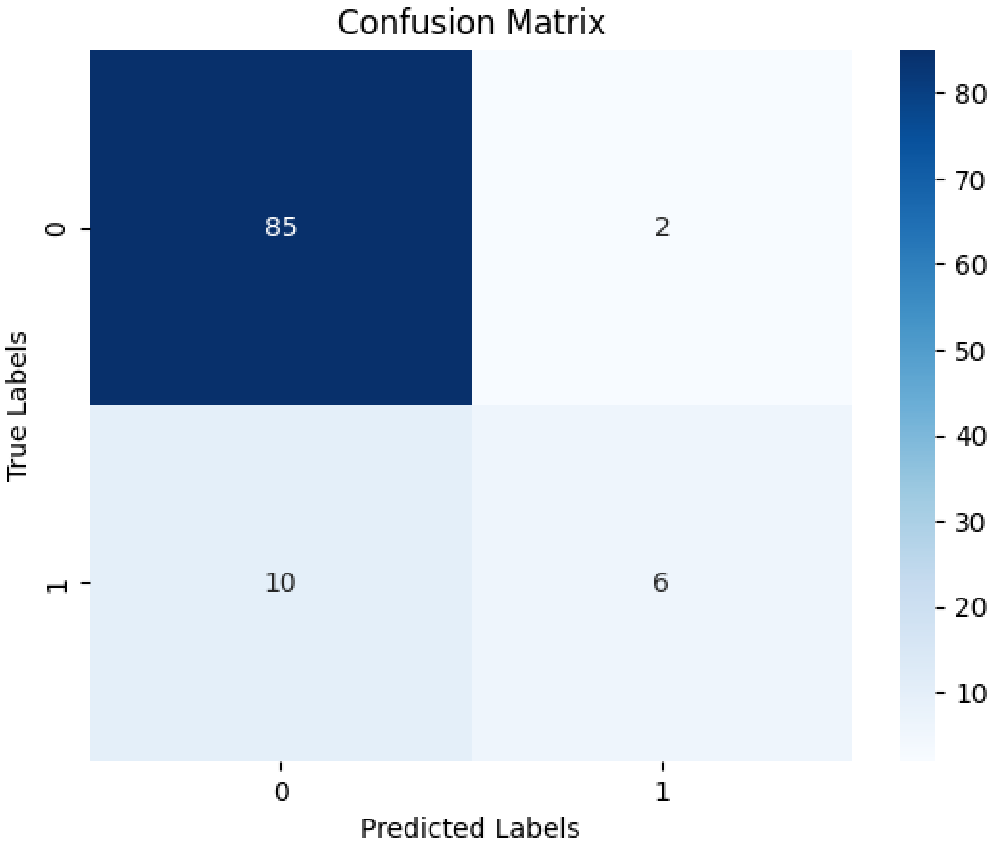
```
f1_score(y_test, prediction1, average = None)
```

[29]:

```
array([0.93406593, 0.5       ])
```

[30]:

```
cm = confusion_matrix(y_true = y_test, y_pred = prediction1)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



```
[31]: from sklearn.neighbors import KNeighborsClassifier
```

```
[32]: #Fitting K-NN to the Training set
      classifier = KNeighborsClassifier(n_neighbors = 3, metric = "minkowski", p = 2)
      classifier.fit(x_train, y_train)
```

```
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\neighbors\_classification.py:239: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
```

```
[32]: KNeighborsClassifier(n_neighbors=3)
```

```
[33]: #Predicting the Test set result
      prediction2 = classifier.predict(x_test)
```

```
[34]: prediction2
```

```
[34]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
              1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
              1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
              1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
              1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
[35]: from sklearn.metrics import confusion_matrix
      from sklearn.metrics import accuracy_score
      confusion_matrix(y_test,prediction2)
```

```
[35]: array([[86,  1],
             [12,  4]], dtype=int64)
```

```
[36]: from sklearn.metrics import precision_score, recall_score, f1_score

      # assuming your predicted and actual labels are stored in variables y_pred and␣
       ↪y_true, respectively
      accuracy = accuracy_score(y_test, prediction2)
      precision = precision_score(y_test, prediction2)
      recall = recall_score(y_test, prediction2)
      f1 = f1_score(y_test, prediction2)

      print("Accuracy:", accuracy)
      print("Precision:", precision)
      print("Recall:", recall)
      print("F1 score:", f1)
```

```
Accuracy: 0.8737864077669902
Precision: 0.8775510204081632
Recall: 0.9885057471264368
F1 score: 0.9297297297297298
```

```
[37]: accuracy_score(y_test,prediction2)
```

```
[37]: 0.8737864077669902
```

```
[38]: probs = Model1.predict_proba(x_test)
      precision_score(y_test, prediction2, average = None)
```

```
[38]: array([0.87755102, 0.8       ])
```

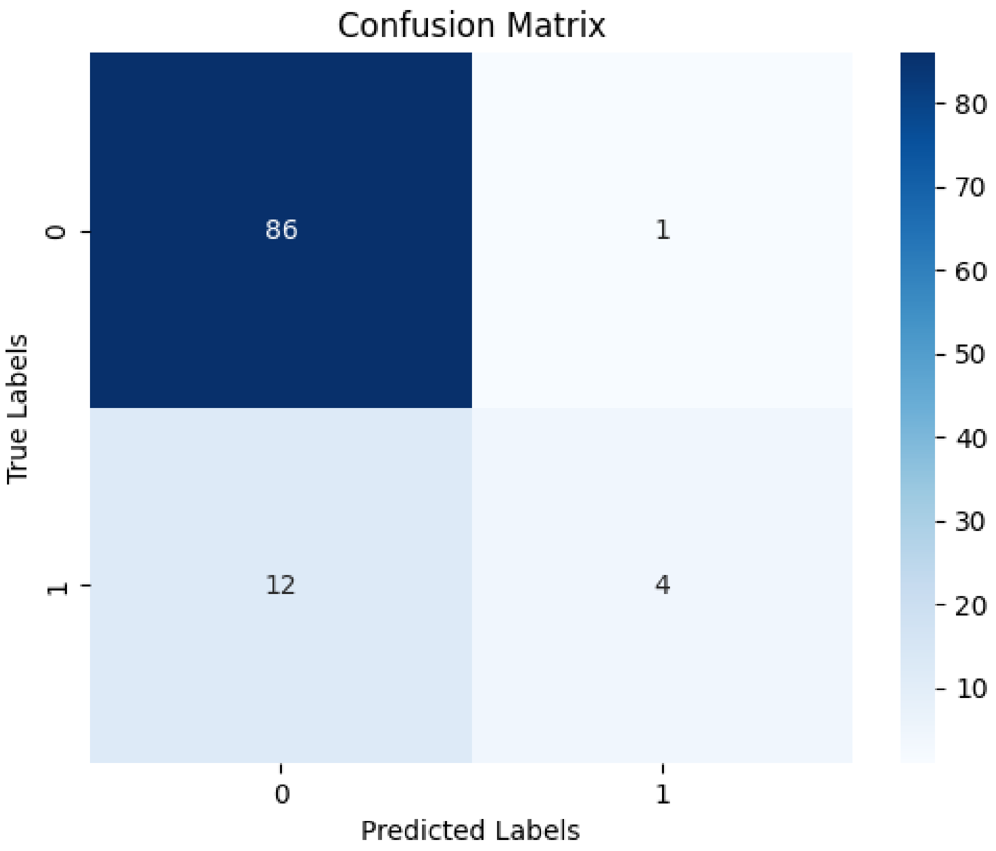
```
[39]: recall_score(y_test, prediction2, average = None)
```

```
[39]: array([0.98850575, 0.25      ])
```

```
[40]: f1_score(y_test, prediction2, average = None)
```

[40]:
```
array([0.92972973, 0.38095238])
```

[41]:
```
cm = confusion_matrix(y_true = y_test, y_pred = prediction2)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



[42]:
```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state = 0,criterion = "entropy")
tree.fit(x_train, y_train)
prediction3 = classifier.predict(x_test)
```

[43]:
```
prediction3
```

[43]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

[44]:
```
confusion_matrix(y_test,prediction3)
```

[44]:
```
array([[86,  1],
       [12,  4]], dtype=int64)
```

[45]:
```
from sklearn.metrics import precision_score, recall_score, f1_score

# assuming your predicted and actual labels are stored in variables y_pred and␣
 ↪y_true, respectively
accuracy = accuracy_score(y_test, prediction3)
precision = precision_score(y_test, prediction3)
recall = recall_score(y_test, prediction3)
f1 = f1_score(y_test, prediction3)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8737864077669902
Precision: 0.8775510204081632
Recall: 0.9885057471264368
F1 score: 0.9297297297297298
```

[46]:
```
accuracy_score(y_test,prediction3)
```

[46]:
```
0.8737864077669902
```

[47]:
```
probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction3, average = None)
```

[47]:
```
array([0.87755102, 0.8       ])
```

[48]:
```
recall_score(y_test, prediction3, average = None)
```

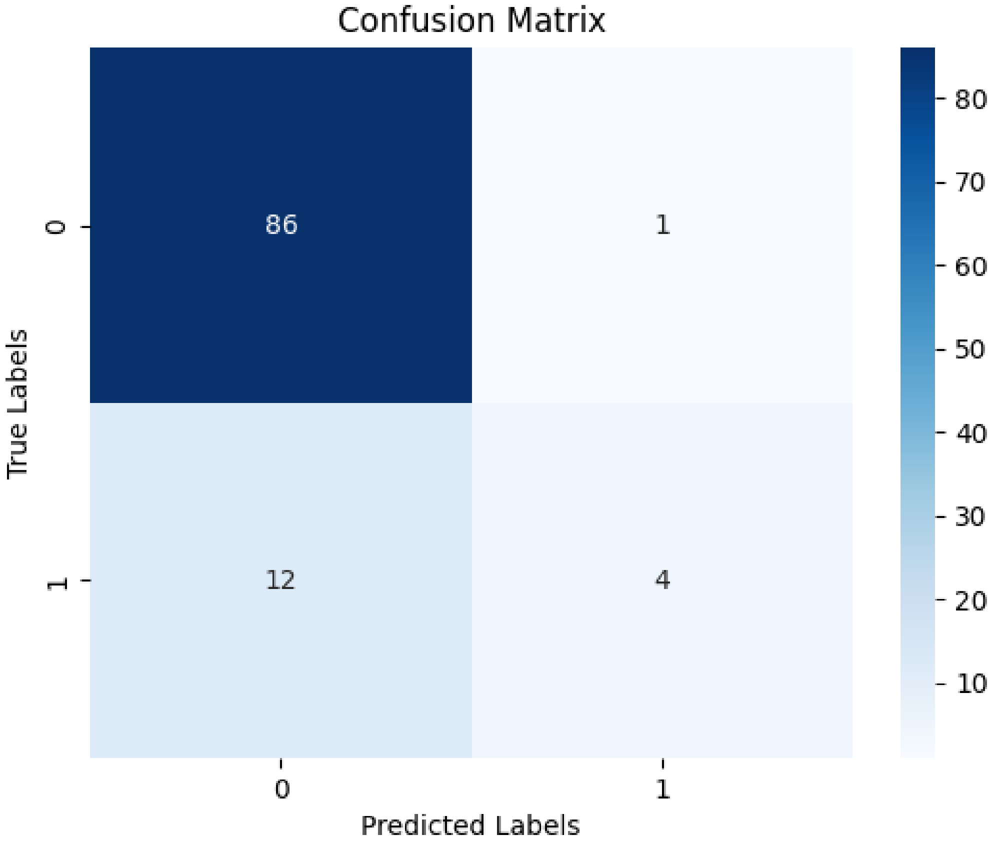
[48]:
```
array([0.98850575, 0.25      ])
```

[49]:
```
f1_score(y_test, prediction3, average = None)
```

[49]:
```
array([0.92972973, 0.38095238])
```

[50]:
```
cm = confusion_matrix(y_true = y_test, y_pred = prediction3)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
```

```
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



[51]:

```
#Support Vector Machine
from sklearn.ensemble import BaggingClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
svm =␣
 ↪OneVsRestClassifier(BaggingClassifier(SVC(C=10,kernel='rbf',random_state=9,probability=True
svm.fit(x_train, y_train)
prediction4 = svm.predict(x_test)
```

[52]:

```
prediction4
```

[52]:

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

```
[53]: confusion_matrix(y_test,prediction4)
```

```
[53]: array([[87,  0],
             [16,  0]], dtype=int64)
```

```
[54]: from sklearn.metrics import precision_score, recall_score, f1_score

      # assuming your predicted and actual labels are stored in variables y_pred and␣
        ↪y_true, respectively
      accuracy = accuracy_score(y_test, prediction4)
      precision = precision_score(y_test, prediction4)
      recall = recall_score(y_test, prediction4)
      f1 = f1_score(y_test, prediction4)

      print("Accuracy:", accuracy)
      print("Precision:", precision)
      print("Recall:", recall)
      print("F1 score:", f1)
```

```
Accuracy: 0.8446601941747572
Precision: 0.8446601941747572
Recall: 1.0
F1 score: 0.9157894736842105
```

```
[55]: accuracy_score(y_test,prediction4)
```

```
[55]: 0.8446601941747572
```

```
[56]: probs = Model1.predict_proba(x_test)
      precision_score(y_test, prediction4, average = None)
```

```
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
[56]: array([0.84466019, 0.        ])
```

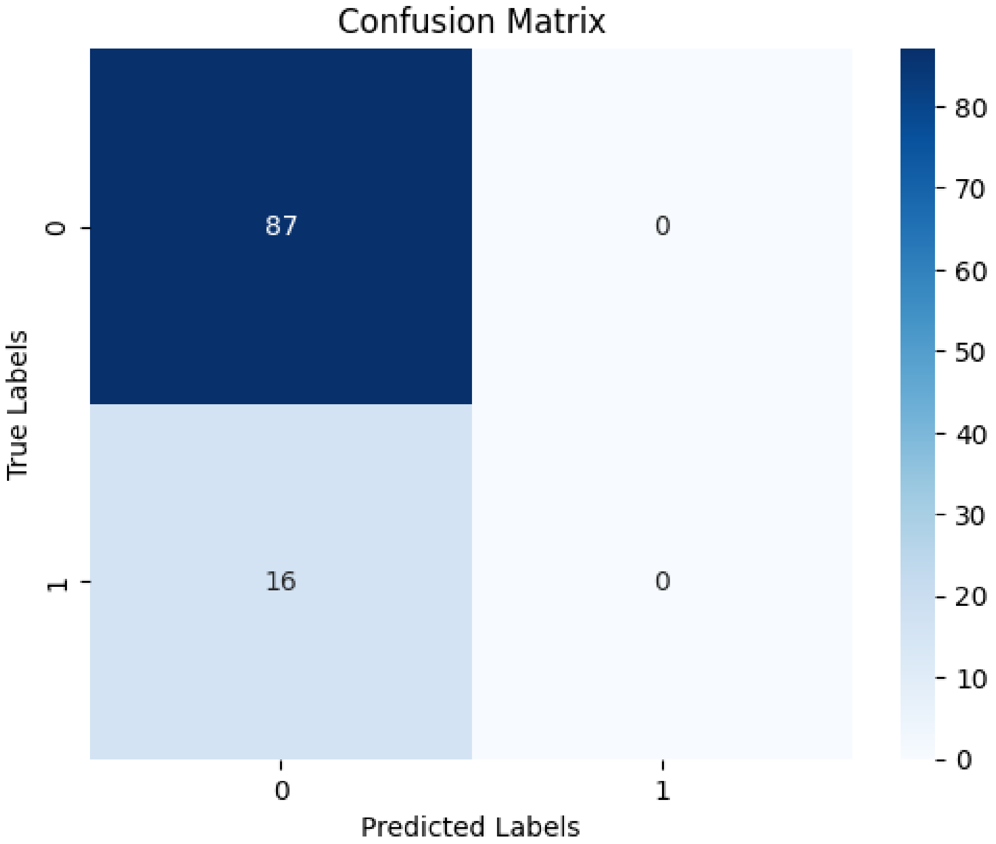
```
[57]: recall_score(y_test, prediction4, average = None)
```

```
[57]: array([1., 0.])
```

[58]:
```
f1_score(y_test, prediction4, average = None)
```

[58]:
```
array([0.91578947, 0.        ])
```

[59]:
```
cm = confusion_matrix(y_true = y_test, y_pred = prediction4)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



[60]:
```
from sklearn.naive_bayes import GaussianNB
nbcla = GaussianNB()
nbcla.fit(x_train, y_train)
prediction5 = nbcla.predict(x_test)
```

```
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\utils\validation.py:1408: DataConversionWarning: A column-
```

```
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[61]:

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
confusion_matrix(y_test,prediction5)
```

[61]:

```
array([[81,  6],
       [ 8,  8]], dtype=int64)
```

[62]:

```
from sklearn.metrics import precision_score, recall_score, f1_score

# assuming your predicted and actual labels are stored in variables y_pred and␣
 ↪y_true, respectively
accuracy = accuracy_score(y_test, prediction5)
precision = precision_score(y_test, prediction5)
recall = recall_score(y_test, prediction5)
f1 = f1_score(y_test, prediction5)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8640776699029126
Precision: 0.9101123595505618
Recall: 0.9310344827586207
F1 score: 0.9204545454545454
```

[63]:

```
accuracy_score(y_test,prediction5)
```

[63]:

```
0.8640776699029126
```

[64]:

```
probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction5, average = None)
```

[64]:

```
array([0.91011236, 0.57142857])
```

[65]:

```
recall_score(y_test, prediction5, average = None)
```

[65]:

```
array([0.93103448, 0.5       ])
```

[66]:

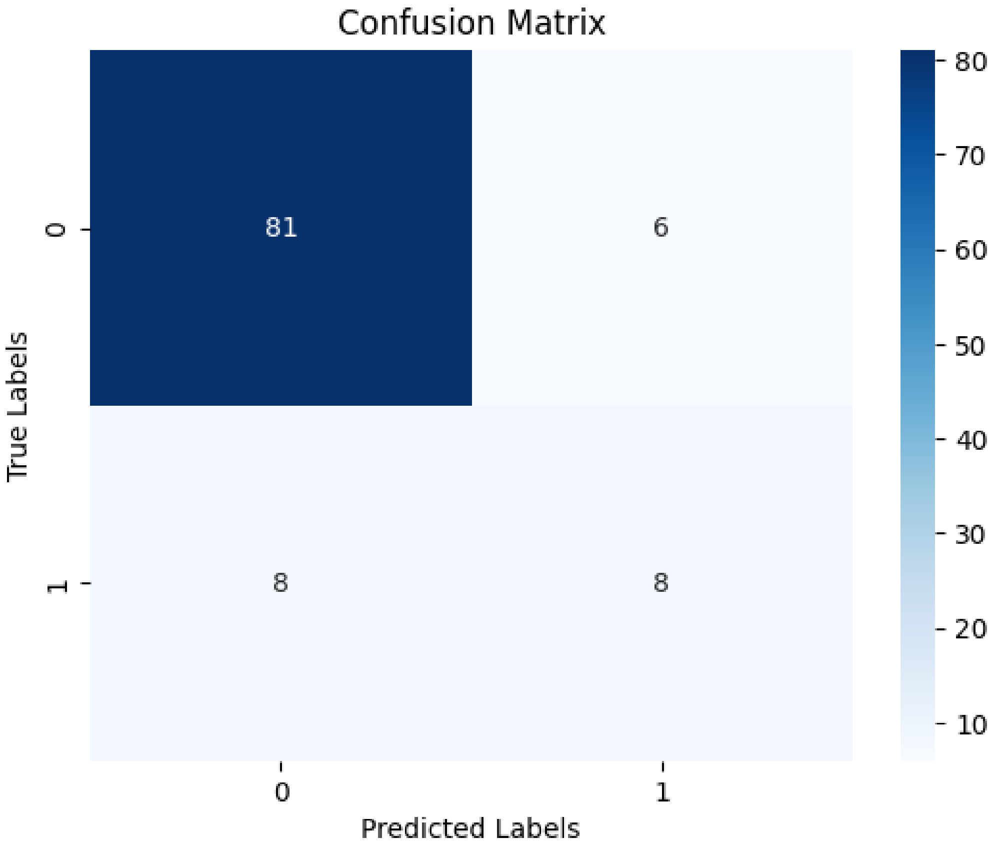
```
f1_score(y_test, prediction5, average = None)
```

[66]:

```
array([0.92045455, 0.53333333])
```

[67]:

```
cm = confusion_matrix(y_true = y_test, y_pred = prediction5)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



[68]:

```
from sklearn.ensemble import RandomForestClassifier

# Initialize the classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model using training dataset
rf_classifier.fit(x_train, y_train)

# Make predictions on test dataset
prediction6 = rf_classifier.predict(x_test)

# Evaluate the accuracy of the model
```

```
#accuracy = rf_classifier.score(x_test, y_test)
#print("Accuracy:", accuracy)
```

```
C:\Users\shrut\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-packages\sklearn\base.py:1389:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using
ravel().
  return fit_method(estimator, *args, **kwargs)
```

[69]:

```
confusion_matrix(y_test,prediction6)
```

[69]:

```
array([[85,  2],
       [ 9,  7]], dtype=int64)
```

[70]:

```
from sklearn.metrics import precision_score, recall_score, f1_score

# assuming your predicted and actual labels are stored in variables y_pred and y_true, respectively
accuracy = accuracy_score(y_test, prediction6)
precision = precision_score(y_test, prediction6)
recall = recall_score(y_test, prediction6)
f1 = f1_score(y_test, prediction6)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8932038834951457
Precision: 0.9042553191489362
Recall: 0.9770114942528736
F1 score: 0.9392265193370166
```

[71]:

```
accuracy_score(y_test,prediction6)
```

[71]:

```
0.8932038834951457
```

[72]:

```
probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction6, average = None)
```

[72]:

```
array([0.90425532, 0.77777778])
```

[73]:

```
recall_score(y_test, prediction6, average = None)
```

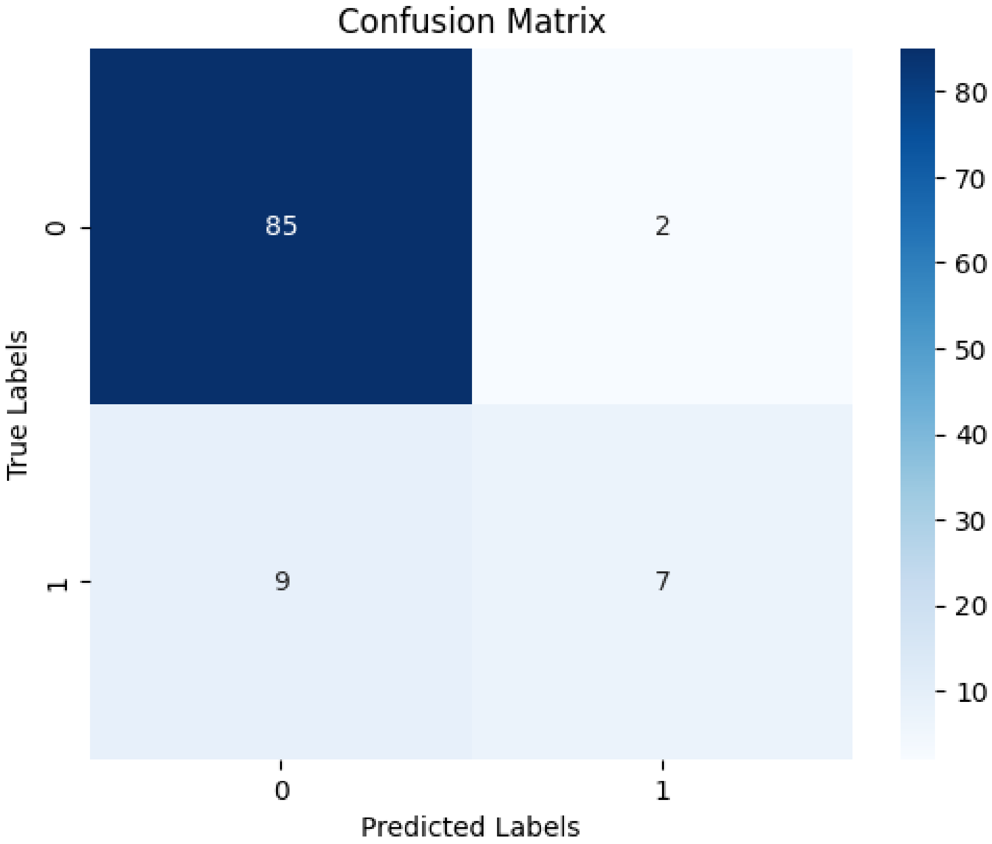
[73]:

```
array([0.97701149, 0.4375    ])
```

[74]:
```
f1_score(y_test, prediction6, average = None)
```

[74]:
```
array([0.93922652, 0.56      ])
```

[75]:
```
cm = confusion_matrix(y_true = y_test, y_pred = prediction6)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



[76]:
```
#Finding Correlation
cn=lung_data.corr()
cn
```

[76]:
```
                   GENDER       AGE   SMOKING  YELLOW_FINGERS   ANXIETY  \
GENDER           1.000000 -0.021306 -0.036277        0.212959  0.152127
AGE             -0.021306  1.000000 -0.084475        0.005205  0.053170
SMOKING         -0.036277 -0.084475  1.000000       -0.014585  0.160267
```

```
YELLOW_FINGERS         0.212959  0.005205 -0.014585        1.000000  0.565829
ANXIETY                0.152127  0.053170  0.160267        0.565829  1.000000
PEER_PRESSURE          0.275564  0.018685 -0.042822        0.323083  0.216841
CHRONIC DISEASE        0.204606 -0.012642 -0.141522        0.041122 -0.009678
FATIGUE                0.083560  0.012614 -0.029575       -0.118058 -0.188538
ALLERGY               -0.154251  0.027990  0.001913       -0.144300 -0.165750
WHEEZING              -0.141207  0.055011 -0.129426       -0.078515 -0.191807
ALCOHOL CONSUMING     -0.454268  0.058985 -0.050623       -0.289025 -0.165750
COUGHING              -0.133303  0.169950 -0.129471       -0.012640 -0.225644
SHORTNESS OF BREATH    0.064911 -0.017513  0.061264       -0.105944 -0.144077
SWALLOWING DIFFICULTY  0.078161 -0.001270  0.030718        0.345904  0.489403
CHEST PAIN            -0.362958 -0.018104  0.120117       -0.104829 -0.113634
LUNG_CANCER            0.067254 -0.089465 -0.058179       -0.181339 -0.144947

                       PEER_PRESSURE  CHRONIC DISEASE   FATIGUE   ALLERGY  \
GENDER                      0.275564         0.204606  0.083560 -0.154251
AGE                         0.018685        -0.012642  0.012614  0.027990
SMOKING                    -0.042822        -0.141522 -0.029575  0.001913
YELLOW_FINGERS              0.323083         0.041122 -0.118058 -0.144300
ANXIETY                     0.216841        -0.009678 -0.188538 -0.165750
PEER_PRESSURE               1.000000         0.048515  0.078148 -0.081800
CHRONIC DISEASE             0.048515         1.000000 -0.110529  0.106386
FATIGUE                     0.078148        -0.110529  1.000000  0.003056
ALLERGY                    -0.081800         0.106386  0.003056  1.000000
WHEEZING                   -0.068771        -0.049967  0.141937  0.173867
ALCOHOL CONSUMING          -0.159973         0.002150 -0.191377  0.344339
COUGHING                   -0.089019        -0.175287  0.146856  0.189524
SHORTNESS OF BREATH        -0.220175        -0.026459  0.441745 -0.030056
SWALLOWING DIFFICULTY       0.366590         0.075176 -0.132790 -0.061508
CHEST PAIN                 -0.094828        -0.036938 -0.010832  0.239433
LUNG_CANCER                -0.186388        -0.110891 -0.150673 -0.327766

                       WHEEZING  ALCOHOL CONSUMING  COUGHING  \
GENDER                -0.141207          -0.454268 -0.133303
AGE                    0.055011           0.058985  0.169950
SMOKING               -0.129426          -0.050623 -0.129471
YELLOW_FINGERS        -0.078515          -0.289025 -0.012640
ANXIETY               -0.191807          -0.165750 -0.225644
PEER_PRESSURE         -0.068771          -0.159973 -0.089019
CHRONIC DISEASE       -0.049967           0.002150 -0.175287
FATIGUE                0.141937          -0.191377  0.146856
ALLERGY                0.173867           0.344339  0.189524
WHEEZING               1.000000           0.265659  0.374265
ALCOHOL CONSUMING      0.265659           1.000000  0.202720
COUGHING               0.374265           0.202720  1.000000
SHORTNESS OF BREATH    0.037834          -0.179416  0.277385
SWALLOWING DIFFICULTY  0.069027          -0.009294 -0.157586
```

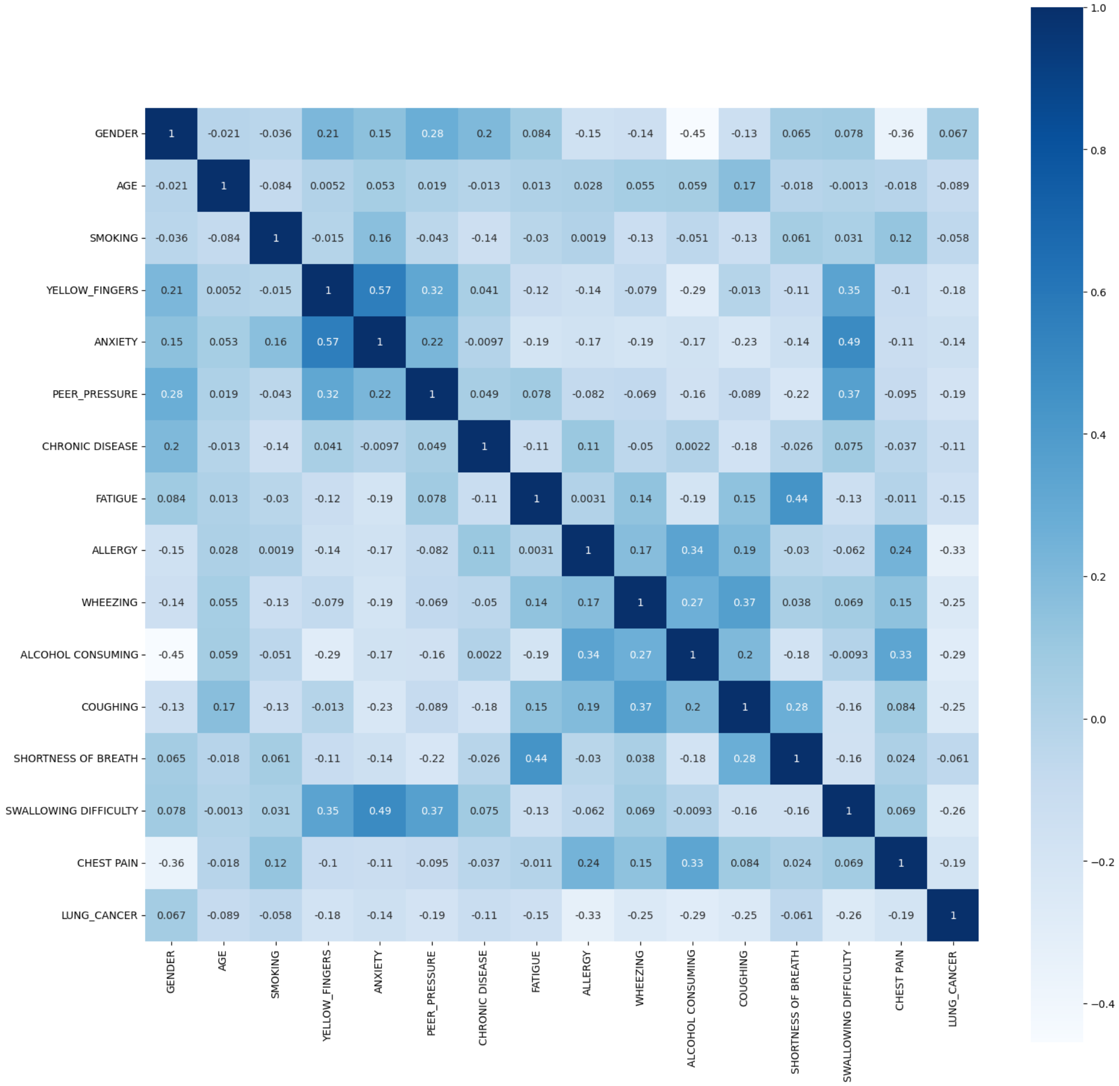
```
CHEST PAIN              0.147640           0.331226  0.083958
LUNG_CANCER            -0.249300          -0.288533 -0.248570

                       SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN  \
GENDER                            0.064911               0.078161   -0.362958
AGE                              -0.017513              -0.001270   -0.018104
SMOKING                           0.061264               0.030718    0.120117
YELLOW_FINGERS                   -0.105944               0.345904   -0.104829
ANXIETY                          -0.144077               0.489403   -0.113634
PEER_PRESSURE                    -0.220175               0.366590   -0.094828
CHRONIC DISEASE                  -0.026459               0.075176   -0.036938
FATIGUE                           0.441745              -0.132790   -0.010832
ALLERGY                          -0.030056              -0.061508    0.239433
WHEEZING                          0.037834               0.069027    0.147640
ALCOHOL CONSUMING                -0.179416              -0.009294    0.331226
COUGHING                          0.277385              -0.157586    0.083958
SHORTNESS OF BREATH               1.000000              -0.161015    0.024256
SWALLOWING DIFFICULTY            -0.161015               1.000000    0.069027
CHEST PAIN                        0.024256               0.069027    1.000000
LUNG_CANCER                      -0.060738              -0.259730   -0.190451

                       LUNG_CANCER
GENDER                    0.067254
AGE                      -0.089465
SMOKING                  -0.058179
YELLOW_FINGERS           -0.181339
ANXIETY                  -0.144947
PEER_PRESSURE            -0.186388
CHRONIC DISEASE          -0.110891
FATIGUE                  -0.150673
ALLERGY                  -0.327766
WHEEZING                 -0.249300
ALCOHOL CONSUMING        -0.288533
COUGHING                 -0.248570
SHORTNESS OF BREATH      -0.060738
SWALLOWING DIFFICULTY    -0.259730
CHEST PAIN               -0.190451
LUNG_CANCER               1.000000
```

[77]:

```
#Correlation
cmap=sns.diverging_palette(260,-10,s=50, l=75, n=6,
as_cmap=True)
plt.subplots(figsize=(18,18))
sns.heatmap(cn,cmap="Blues",annot=True, square=True)
plt.show()
```



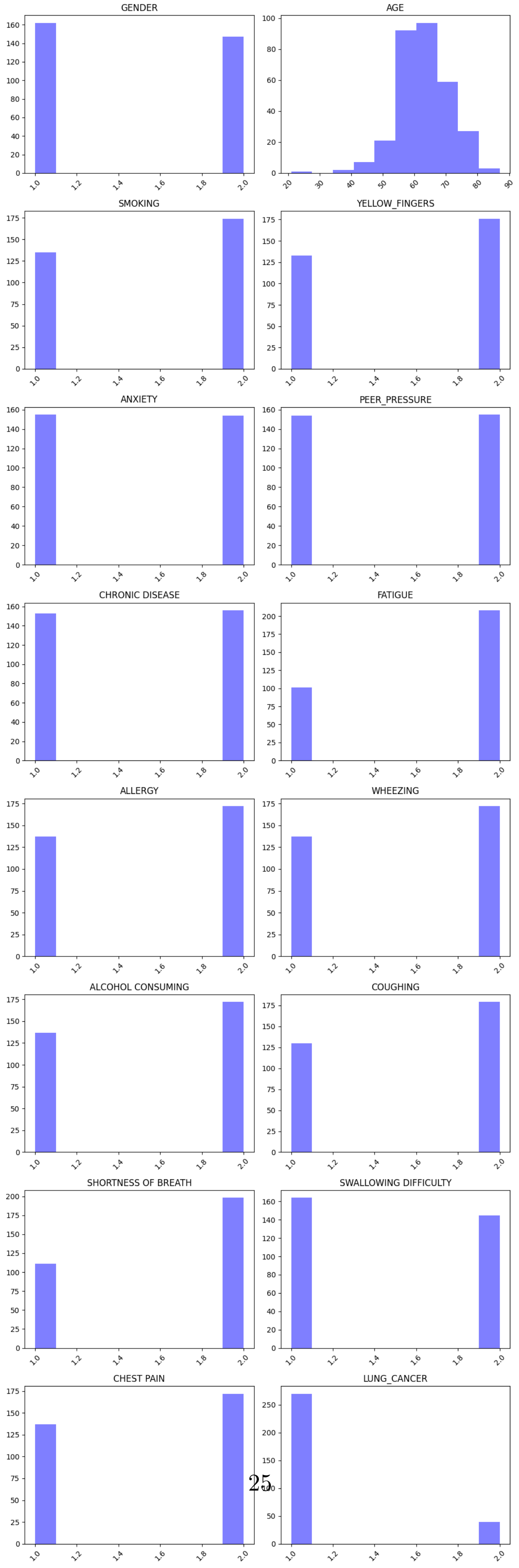
[78]:

```
num_list = list(lung_data.columns)

fig = plt.figure(figsize=(10,30))

for i in range(len(num_list)):
    plt.subplot(8,2,i+1)
    plt.title(num_list[i])
    plt.xticks(rotation=45)
    plt.hist(lung_data[num_list[i]],color='blue',alpha=0.5)

plt.tight_layout()
```


GENDER
AGE
SMOKING
YELLOW_FINGERS
ANXIETY
PEER_PRESSURE
CHRONIC DISEASE
FATIGUE
ALLERGY
WHEEZING
ALCOHOL CONSUMING
COUGHING
SHORTNESS OF BREATH
SWALLOWING DIFFICULTY
CHEST PAIN
LUNG_CANCER
1.0
1.2
1.4
1.6
1.8
2.0
20
30
40
50
60
70
80
90
0
20
40
60
80
100
120
140
160
25
50
75
100
125
150
175
200
250

[ ]:

```
```