In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv(r"survey lung cancer.csv")
df.head()
```

Out[2]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	I
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 [3]:
```
df.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 [4]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
#converting string labels into numbers
df.GENDER = le.fit_transform(df.GENDER)
df.LUNG_CANCER = le.fit_transform(df.LUNG_CANCER)
```

In [5]:
```
df.head()
```

Out[5]:

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

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
33
```

In [7]:
```
dfy = df.LUNG_CANCER==1
```

In [8]:
```
dfy.info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 309 entries, 0 to 308
Series name: LUNG_CANCER
Non-Null Count  Dtype
--------------  -----
309 non-null    bool
dtypes: bool(1)
memory usage: 437.0 bytes
```

In [9]:
```
df=df.drop_duplicates(keep='first')
```

In [10]:
```
df.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
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 [12]:
```
Linput = df.drop(["LUNG_CANCER","PEER_PRESSURE"],axis = 1) #we dropped pee
```

In [13]:
```
Linput.head()
```

Out[13]:

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

In [14]:
```
from sklearn.model_selection import train_test_split
```

In [15]:
```
x_train, x_test, y_train,y_test =train_test_split(Linput, df.LUNG_CANCER,
```

In [16]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [17]:
```
knnmodel=KNeighborsClassifier(n_neighbors=5)
```

In [18]:
```
knnmodel.fit(x_train, y_train)
```

Out[18]:
```
KNeighborsClassifier()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]:
```
#Giving Sample data for prediction
htest=[[1,22,1,2,1,1,1,1,1,1,2,1,1]]
htest=pd.DataFrame(htest , columns=['GENDER','AGE','SMOKING','YELLOW_FINGE
htest
```

Out[19]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	CHRONIC DISEASE	FATIGUE	ALLERGY
0	1	22	1	2	1	1	1	1

In [20]:
```
knnmodel.predict(htest)
```

Out[20]:
```
array([1])
```

In [21]:
```
y_predictK = knnmodel.predict(x_test)
from sklearn import metrics
print("Accuracy:", metrics.accuracy_score(y_test, y_predictK))
```

```
Accuracy: 0.8928571428571429
```

In [22]:
```
from sklearn.naive_bayes import MultinomialNB # as the dataset is discrete
Model= MultinomialNB()
```

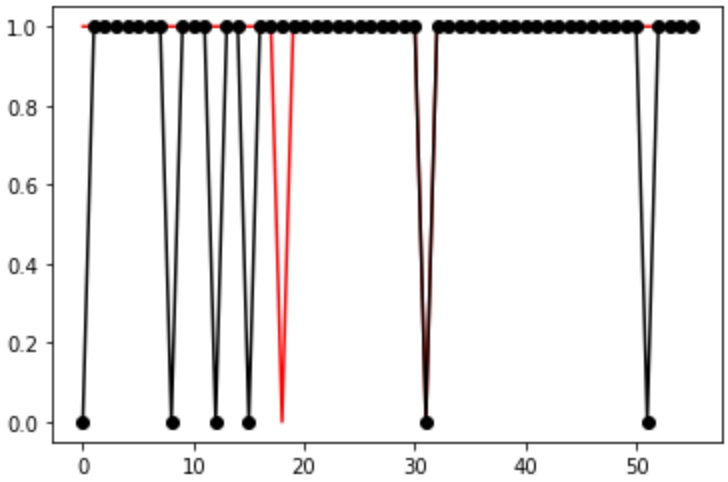
In [23]:
```
Model.fit(x_train, y_train)
#predict response for test dataset
y_predict = Model.predict(x_test)
```

In [24]:
```
from sklearn import metrics
print("Accuracy:", metrics.accuracy_score(y_test, y_predict))
```

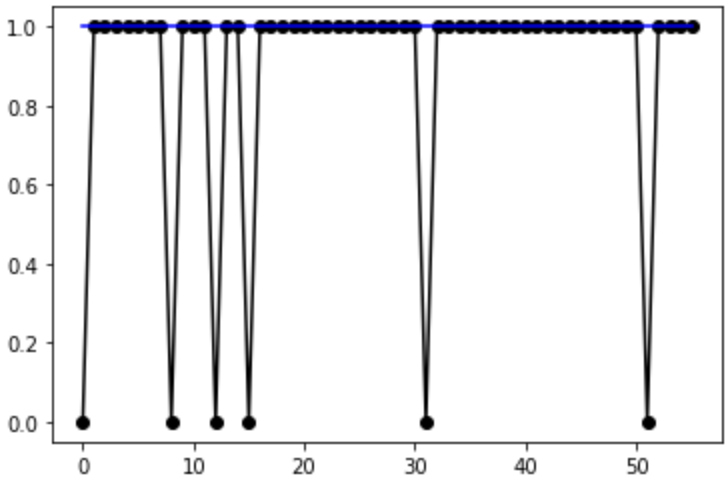
```
Accuracy: 0.8928571428571429
```

In [25]:
```
s = list(y_test)
```

In [26]:
```
import matplotlib.pyplot as plt
plt.plot(y_predictK , color = 'red') #KNN
plt.plot(s , marker = 'o' , color = 'black')
plt.show()
```



In [27]:
```
plt.plot(s , marker = 'o' , color = 'black')
plt.plot(y_predict , color = 'blue')# Naive
plt.show()
```



In [ ]:

In [ ]: