K.SASI KIRAN

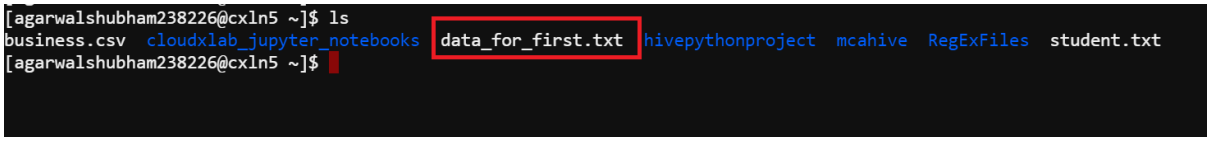
MCA(R)

2019202049

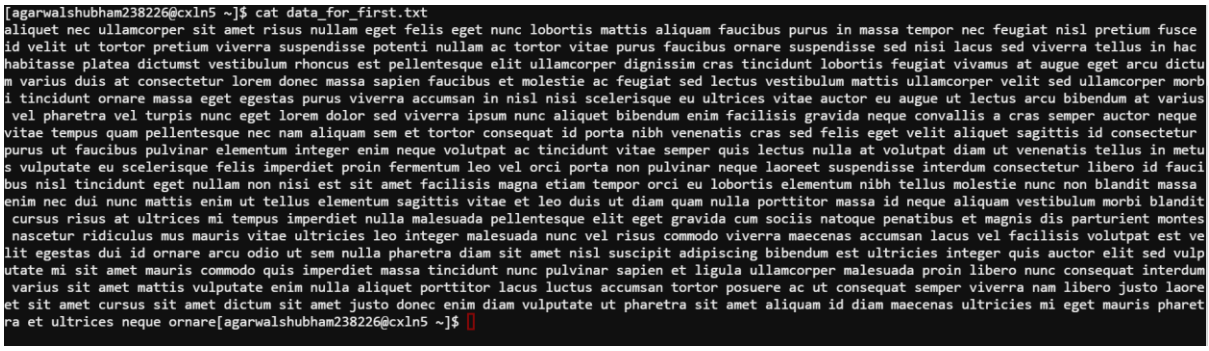
QUESTION 1:

IN HADOOP

TXT file.



File contains words with spaces.



Output 1(a).

Converting the text file by including comma (,) instead of space:

```
[agarwalshubham238226@cxln5 ~]$ hadoop fs -cat data_for_first.txt | sed 's/ /,/g' |more
aliquet,nec,ullamcorper,sit,amet,risus,nullam,eget,felis,eget,nunc,lobortis,mattis,aliquam,faucibus,purus,in,massa,tempor,nec,feugiat,nisl,pretium,fusce,
id,velit,ut,tortor,pretium,viverra,suspendisse,potenti,nullam,ac,tortor,vitae,purus,faucibus,ornare,suspendisse,sed,nisi,lacus,sed,viverra,tellus,in,hac,
habitasse,platea,dictumst,vestibulum,rhoncus,est,pellentesque,elit,ullamcorper,dignissim,cras,tincidunt,lobortis,feugiat,vivamus,at,augue,eget,arcu,dictu
m,varius,duis,at,consectetur,lorem,donec,massa,sapien,faucibus,et,molestie,ac,feugiat,sed,lectus,vestibulum,mattis,ullamcorper,velit,sed,ullamcorper,morb
i,tincidunt,ornare,massa,eget,egestas,purus,viverra,accumsan,in,nisl,nisi,scelerisque,eu,ultrices,vitae,auctor,eu,augue,ut,lectus,arcu,bibendum,at,varius
,vel,pharetra,vel,turpis,nunc,eget,lorem,dolor,sed,viverra,ipsum,nunc,aliquet,bibendum,enim,facilisis,gravida,neque,convallis,a,cras,semper,auctor,neque,
vitae,tempus,quam,pellentesque,nec,nam,aliquam,sem,et,tortor,consequat,id,porta,nibh,venenatis,cras,sed,felis,eget,velit,aliquet,sagittis,id,consectetur,
purus,ut,faucibus,pulvinar,elementum,integer,enim,neque,volutpat,ac,tincidunt,vitae,semper,quis,lectus,nulla,at,volutpat,diam,ut,venenatis,tellus,in,metu
s,vulputate,eu,scelerisque,felis,imperdiet,proin,fermentum,leo,vel,orci,porta,non,pulvinar,neque,laoreet,suspendisse,interdum,consectetur,libero,id,fauci
bus,nisl,tincidunt,eget,nullam,non,nisi,est,sit,amet,facilisis,magna,etiam,tempor,orci,eu,lobortis,elementum,nibh,tellus,molestie,nunc,non,blandit,massa,
enim,nec,dui,nunc,mattis,enim,ut,tellus,elementum,sagittis,vitae,et,leo,duis,ut,diam,quam,nulla,porttitor,massa,id,neque,aliquam,vestibulum,morbi,blandit
,cursus,risus,at,ultrices,mi,tempus,imperdiet,nulla,malesuada,pellentesque,elit,eget,gravida,cum,sociis,natoque,penatibus,et,magnis,dis,parturient,montes
,nascetur,ridiculus,mus,mauris,vitae,ultricies,leo,integer,malesuada,nunc,vel,risus,commodo,viverra,maecenas,accumsan,lacus,vel,facilisis,volutpat,est,ve
lit,egestas,dui,id,ornare,arcu,odio,ut,sem,nulla,pharetra,diam,sit,amet,nisl,suscipit,adipiscing,bibendum,est,ultricies,integer,quis,auctor,elit,sed,vulp
utate,mi,sit,amet,mauris,commodo,quis,imperdiet,massa,tincidunt,nunc,pulvinar,sapien,et,ligula,ullamcorper,malesuada,proin,libero,nunc,consequat,interdum
,varius,sit,amet,mattis,vulputate,enim,nulla,aliquet,porttitor,lacus,luctus,accumsan,tortor,posuere,ac,ut,consequat,semper,viverra,nam,libero,justo,laore
et,sit,amet,cursus,sit,amet,dictum,sit,amet,justo,donec,enim,diam,vulputate,ut,pharetra,sit,amet,aliquam,id,diam,maecenas,ultricies,mi,eget,mauris,pharet
ra,et,ultrices,neque,ornare
```

Now, replacing the content of file with this comma separated data:

```
[agarwalshubham238226@cxln5 ~]$
[agarwalshubham238226@cxln5 ~]$ hadoop fs -cat data_for_first.txt | sed 's/ /,/g' |more > data_for_first.txt
[agarwalshubham238226@cxln5 ~]$
```

Code for mapper :

```
[agarwalshubham238226@cxln5 ~]$ cat mapper.py
import sys

for line in sys.stdin:
    line = line.strip()

    words = line.split()

    for word in words:
        print '%s\t%s' % (word, 1)

[agarwalshubham238226@cxln5 ~]$
```

Code for reducer :

```
[agarwalshubham238226@cxln5 ~]$ cat reducer.py
from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:

    line = line.strip()

    word, count = line.split('\t', 1)

    try:
        count = int(count)
    except ValueError:
        continue

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

if current_word == word:
    print '%s\t%s' % (current_word, current_count)
```

Running mapper and reducer under Hadoop streaming jar:

```
[agarwalshubham238226@cxln5 ~]$ hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \
> -file /home/agarwalshubham238226/mapper.py    -mapper /home/agarwalshubham238226/mapper.py \
> -file /home/agarwalshubham238226/reducer.py   -reducer /home/agarwalshubham238226/reducer.py \
> -input /home/agarwalshubham238226/data_for_first.txt -output /home/agarwalshubham238226/first-output
21/05/20 09:11:15 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/home/agarwalshubham238226/mapper.py, /home/agarwalshubham238226/reducer.py] [/usr/hdp/2.6.2.0-205/hadoop-mapreduce/hadoop-streaming-2
3.2.6.2.0-205.jar] /tmp/streamjob2804730577827721105.jar tmpDir=null
21/05/20 09:11:16 INFO client.RMProxy: Connecting to ResourceManager at cxln2.c.thelab-240901.internal/10.142.1.2:8050
21/05/20 09:11:16 INFO client.AHSProxy: Connecting to Application History server at cxln2.c.thelab-240901.internal/10.142.1.2:10200
21/05/20 09:11:16 INFO client.RMProxy: Connecting to ResourceManager at cxln2.c.thelab-240901.internal/10.142.1.2:8050
21/05/20 09:11:16 INFO client.AHSProxy: Connecting to Application History server at cxln2.c.thelab-240901.internal/10.142.1.2:10200
21/05/20 09:11:17 INFO mapreduce.JobSubmitter: Cleaning up the staging area /user/agarwalshubham238226/.staging/job_1621159569377_0761
```

Output 1(b):

```
[agarwalshubham238226@cxln5 ~]$ cat data_for_first.txt | /home/agarwalshubham238226/mapper.py
aliquet 1
nec     1
ullamcorper     1
sit     1
amet    1
risus   1
nullam  1
eget    1
felis   1
eget    1
nunc    1
lobortis        1
mattis  1
aliquam 1
faucibus        1
purus   1
in      1
massa   1
tempor  1
nec     1
feugiat 1
```

Output 1(c):

```
[agarwalshubham238226@cxln5 ~]$ cat data_for_first.txt | /home/agarwalshubham238226/mapper.py | sort -k1,1 | /home/agarwalshubham238226/reducer.py
a       1
ac      4
accumsan        3
adipiscing      1
aliquam 4
aliquet 4
amet    9
arcu    3
at      5
auctor  3
augue   2
bibendum        3
blandit 2
commodo 2
consectetur     3
consequat       3
convallis       1
cras    3
cum     1
cursus  2
diam    5
dictum  2
dictumst        1
dignissim       1
dis     1
dolor   1
donec   2
dui     2
duis    2
egestas 2
eget    9
```

Output 1(d):

```
[agarwalshubham238226@cxln5 ~]$ hadoop fs -cat data_for_first.txt | grep -io '\<[aeiou]*\>'
eu
eu
a
eu
eu
```

```
[agarwalshubham238226@cxln5 ~]$ hadoop fs -cat data_for_first.txt | grep -io '\<[aeiou]*\>' | wc
      5       5      14
```

K SASI KIRAN

MCA(R)

2019202049

QUESTION 2

In [1]:
```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
```

2A

Load data

In [3]:
```
df = pd.read_csv('heart.csv')
df.head(5)
```

Out[3]:

	Patient ID	Patient Name	Age	Phone	Disease
0	1	Sakshi	34	8901782367	Disease1
1	2	Madhu	45	9089876715	Disease1
2	3	Ganesh	30	8989889898	Disease2
3	4	Kumar	20	8767564534	Disease1
4	5	Mani	19	9101918171	Disease3

2B

SELECT FEATURES

In [4]:
```
nRow, nCol = df.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 15 rows and 5 columns
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Patient ID', 'Patient Name', 'Age', 'Phone', 'Disease'], dtype='objec
t')
```

In [6]:
```
df.head()
```

Out[6]:

	Patient ID	Patient Name	Age	Phone	Disease
0	1	Sakshi	34	8901782367	Disease1
1	2	Madhu	45	9089876715	Disease1
2	3	Ganesh	30	8989889898	Disease2
3	4	Kumar	20	8767564534	Disease1
4	5	Mani	19	9101918171	Disease3

2C

TRAINING AND TESTING DATASET

In [8]:
```
x = df[["Patient ID", "Age"]]
y = df["Patient Name"]

x_train,x_text,y_train,y_test = train_test_split(x,y, test_size=0.3, random_state
x_train
```

Out[8]:

	Patient ID	Age
4	5	19
1	2	45
13	14	57
0	1	34
14	15	37
9	10	29
8	9	44
12	13	55
11	12	25
5	6	15

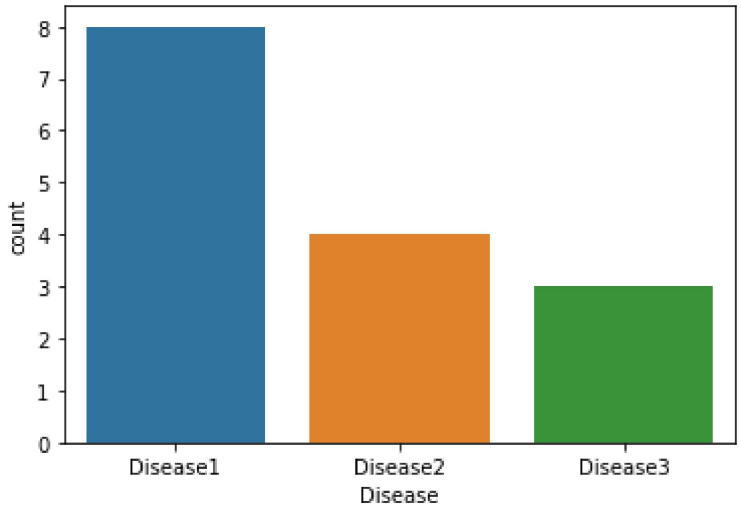
2D

LOGISTIC REGRESSION

In [18]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import math
```

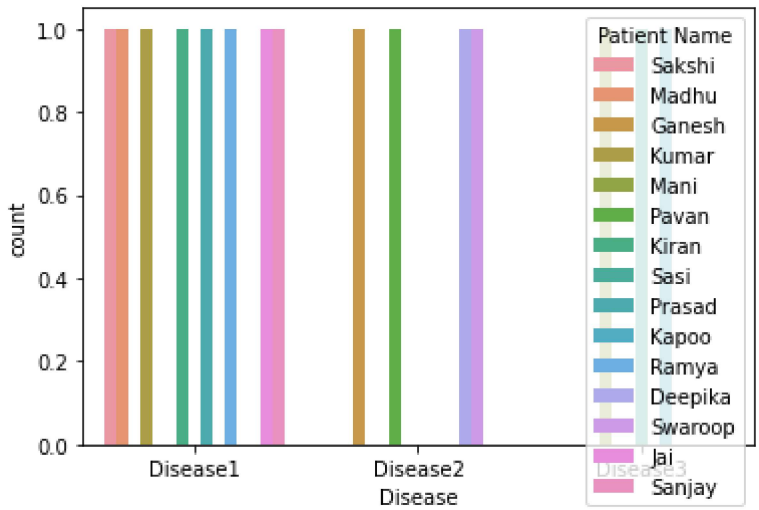
In [19]:
```
sns.countplot(x="Disease",data=df)
```

Out[19]: <AxesSubplot:xlabel='Disease', ylabel='count'>



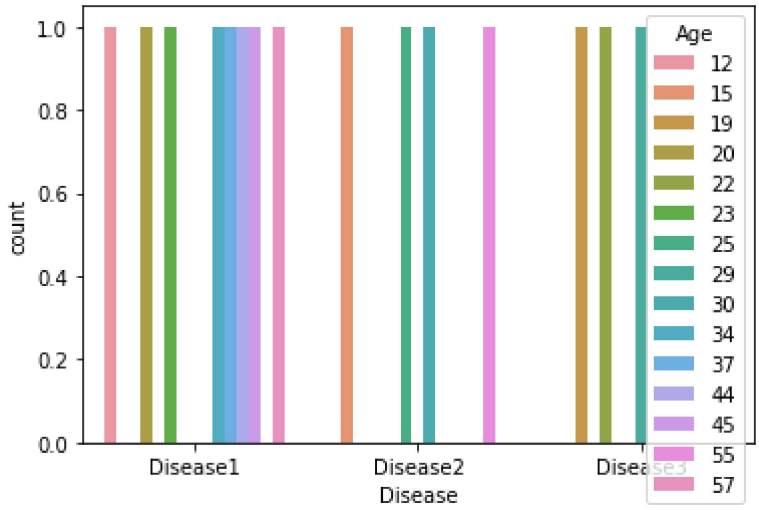
In [21]:
```
sns.countplot(x="Disease",hue="Patient Name",data=df)
```

Out[21]: <AxesSubplot:xlabel='Disease', ylabel='count'>



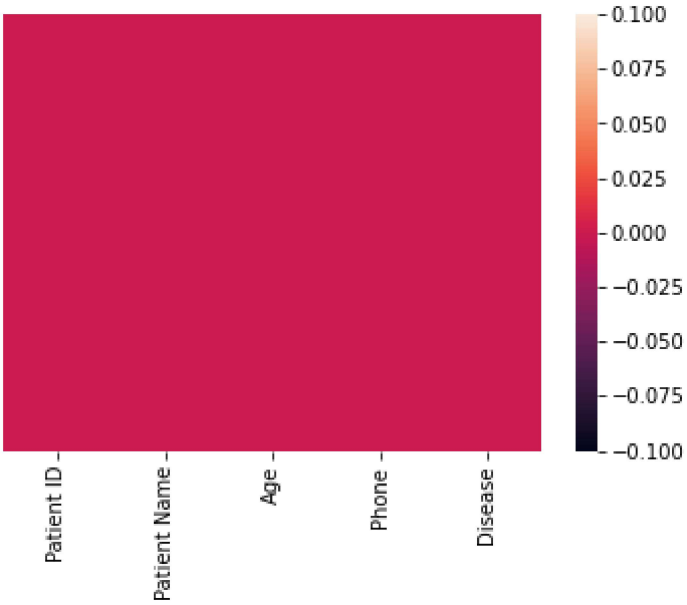
In [22]:
```
sns.countplot(x="Disease",hue="Age",data=df)
```

Out[22]: <AxesSubplot:xlabel='Disease', ylabel='count'>



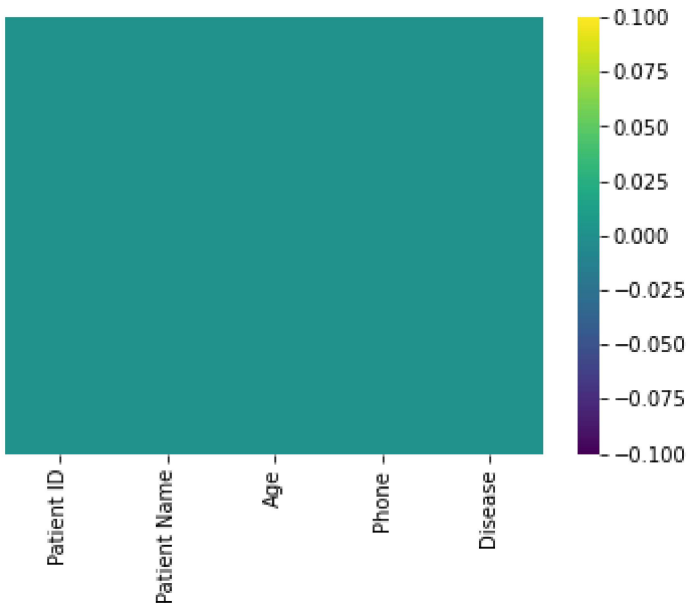
In [23]:
```
sns.heatmap(df.isnull() , yticklabels=False)
```

Out[23]: <AxesSubplot:>



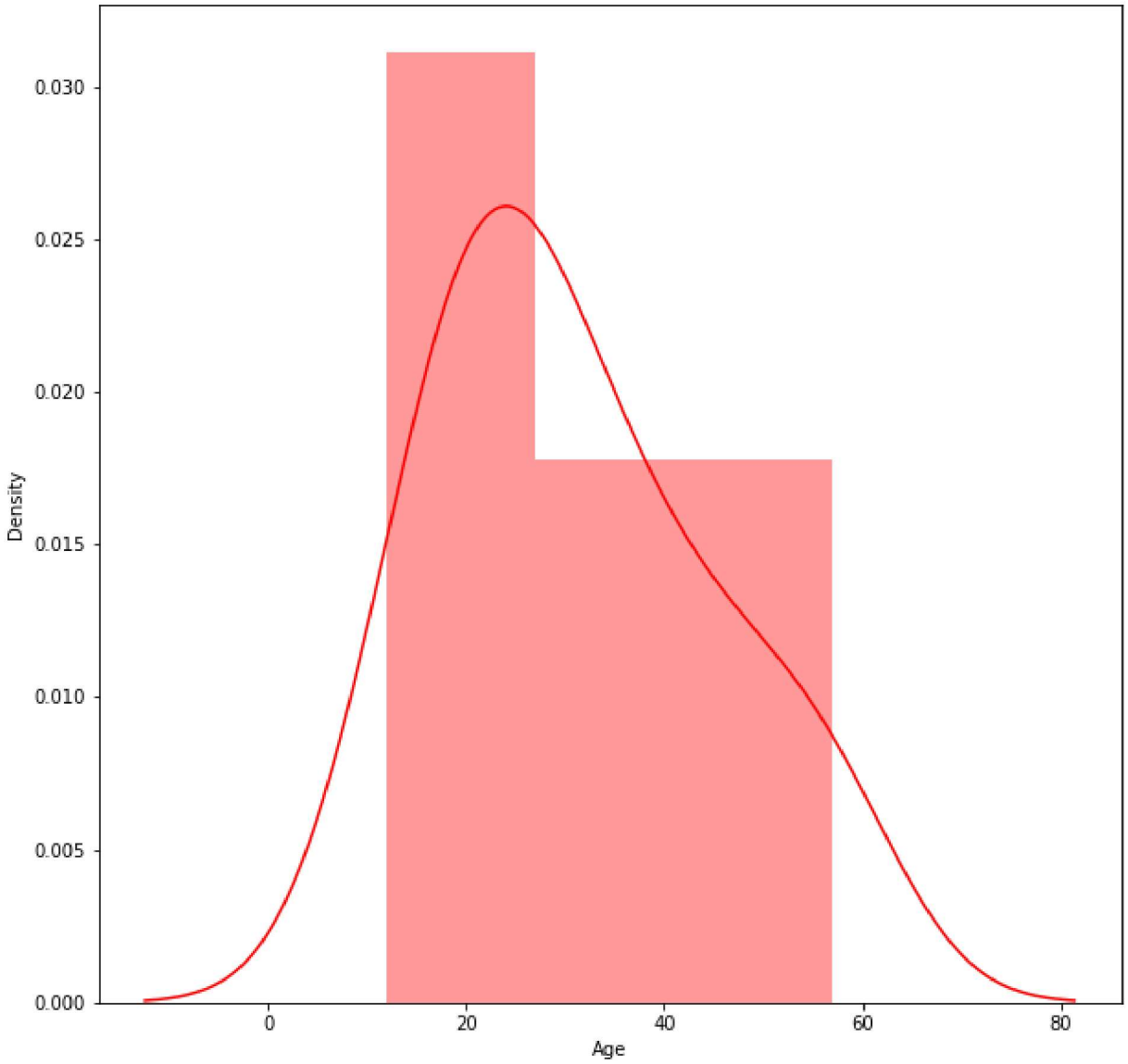
In [24]:
```
sns.heatmap(df.isnull() , yticklabels=False , cmap="viridis")
```

Out[24]: <AxesSubplot:>



In [26]:
```
plt.figure(figsize = (10, 10))
sns.distplot(df['Age'], color = 'red')
```

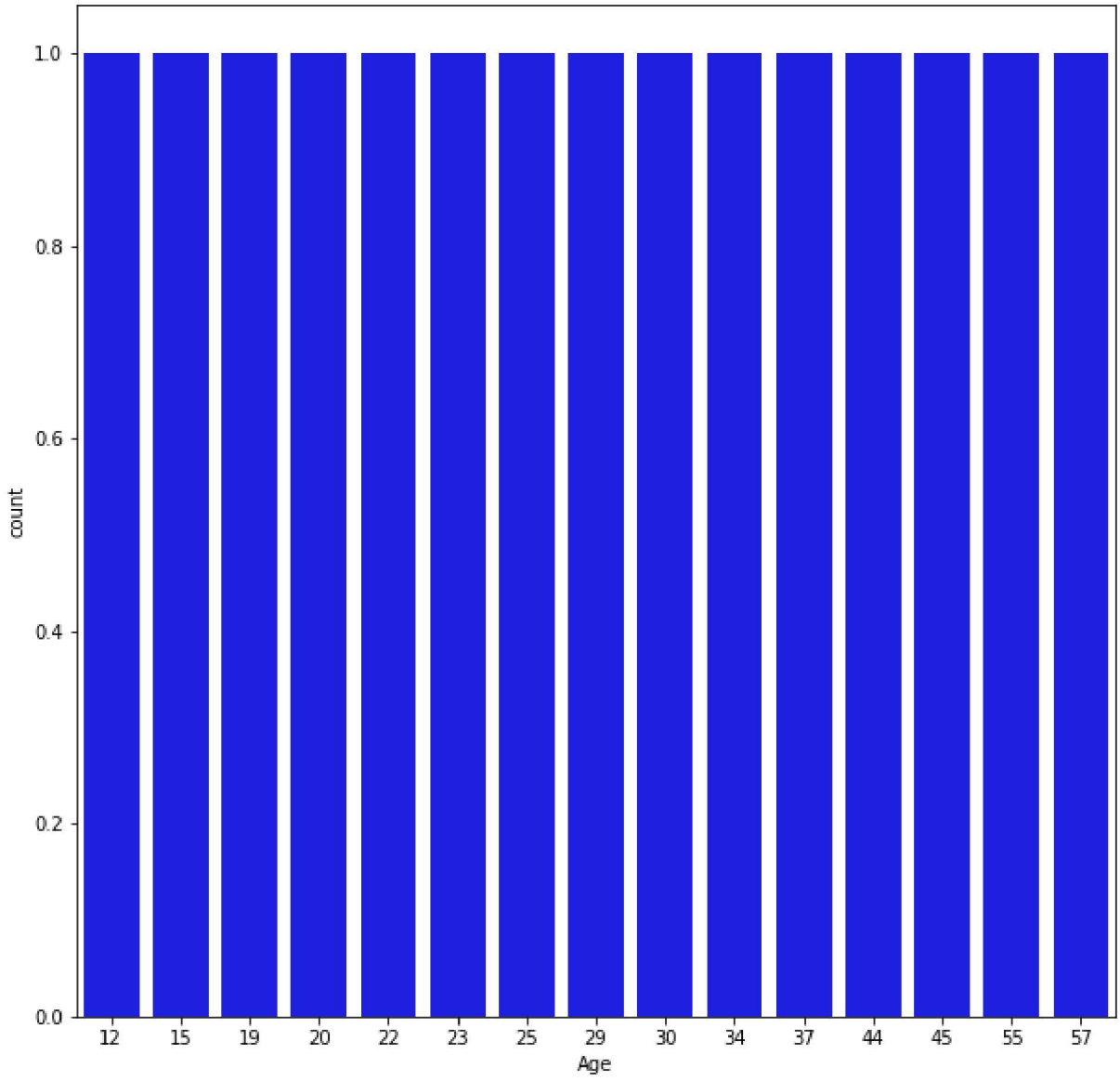
Out[26]: <AxesSubplot:xlabel='Age', ylabel='Density'>



In [27]:
```
plt.figure(figsize = (10, 10))
sns.countplot(df['Age'], color = 'blue')
```

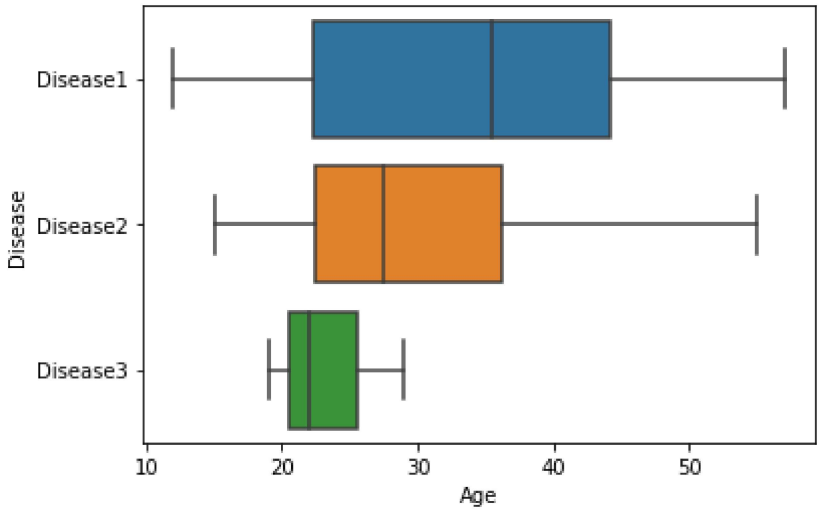
```
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\seabo
rn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, an
d passing other arguments without an explicit keyword will result in an error o
r misinterpretation.
  warnings.warn(
```

Out[27]: <AxesSubplot:xlabel='Age', ylabel='count'>



In [28]:
```
sns.boxplot(x = 'Age', y = 'Disease', data = df)
```

Out[28]:
```
<AxesSubplot:xlabel='Age', ylabel='Disease'>
```



2e

CONFUSION MATRIX

In [35]:
```
model = SVC(gamma=2, C=0.7)
```

In [37]:
```
model.fit(x_train,y_train)
```

Out[37]:
```
SVC(C=0.7, gamma=2)
```

In [39]:
```
y_pred = model.predict(x_text)
```

In [40]:
```
acc = metrics.accuracy_score(y_pred,y_test)
```

In [42]:
```
cnf_mat = metrics.confusion_matrix(y_test, y_pred)
cnf_mat
```

Out[42]:
```
array([[0, 0, 0, 0, 0, 0, 1, 0],
       [0, 0, 0, 1, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 0, 0, 0]], dtype=int64)
```

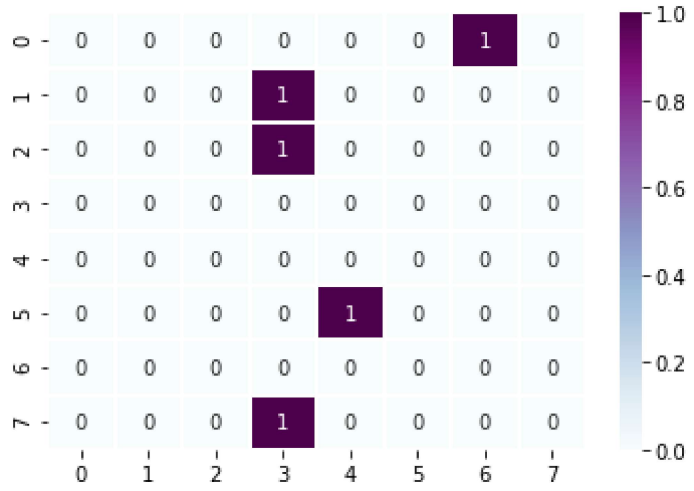
In [44]:
```
rep = metrics.classification_report(y_test, y_pred)
print(rep)
```

```
              precision    recall  f1-score   support

      Ganesh       0.00      0.00      0.00       1.0
       Kiran       0.00      0.00      0.00       1.0
       Kumar       0.00      0.00      0.00       1.0
        Mani       0.00      0.00      0.00       0.0
       Pavan       0.00      0.00      0.00       0.0
       Ramya       0.00      0.00      0.00       1.0
      Sakshi       0.00      0.00      0.00       0.0
        Sasi       0.00      0.00      0.00       1.0

    accuracy                           0.00       5.0
   macro avg       0.00      0.00      0.00       5.0
weighted avg       0.00      0.00      0.00       5.0


c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Us
e `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Recall and F-score
are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Us
e `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Recall and F-score
are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Us
e `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\users\sasi\appdata\local\programs\python\python38-32\lib\site-packages\sklea
rn\metrics\_classification.py:1245: UndefinedMetricWarning: Recall and F-score
are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_
division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

QUESTION 3

In [46]:
```
import seaborn, numpy as np
metrics.accuracy_score(y_test, y_pred)
seaborn.heatmap(cnf_mat,linewidths=1,annot=True,cmap="BuPu")
```

Out[46]: <AxesSubplot:>



In [49]:
```
age = []
for i in range (0,len (df ['Age'])):
    if 10 < df ['Age'][i] <=20:
        age.append('Young');
    elif 20 < df['Age'][i] <=40:
        age.append('Middle');
    else:
        age.append('Old');

age_data = pd.DataFrame(data = age, columns = ['Age_category'])
augmented_data = pd.concat([df, age_data], axis=1)
augmented_data.head()
```

Out[49]:

	Patient ID	Patient Name	Age	Phone	Disease	Age_category
0	1	Sakshi	34	8901782367	Disease1	Middle
1	2	Madhu	45	9089876715	Disease1	Old
2	3	Ganesh	30	8989889898	Disease2	Middle
3	4	Kumar	20	8767564534	Disease1	Young
4	5	Mani	19	9101918171	Disease3	Young

In [50]:
```
augmented_data.to_excel('Ages.xlsx')
```

In [51]:
```
df = pd.read_excel("Ages.xlsx")
print(df.head)
```

```
<bound method NDFrame.head of     Unnamed: 0  Patient ID Patient Name  Age
Phone   Disease  \
0            0           1       Sakshi   34  8901782367  Disease1
1            1           2        Madhu   45  9089876715  Disease1
2            2           3       Ganesh   30  8989889898  Disease2
3            3           4        Kumar   20  8767564534  Disease1
4            4           5         Mani   19  9101918171  Disease3
5            5           6        Pavan   15  8767564534  Disease2
6            6           7        Kiran   23  8167156545  Disease1
7            7           8         Sasi   22  9089785613  Disease3
8            8           9       Prasad   44  9089123456  Disease1
9            9          10        Kapoo   29  7867564534  Disease3
10          10          11        Ramya   12  6756458989  Disease1
11          11          12      Deepika   25  7867564534  Disease2
12          12          13      Swaroop   55  8978671234  Disease2
13          13          14          Jai   57  9012345567  Disease1
14          14          15       Sanjay   37  9012345890  Disease1

   Age_category
0        Middle
1           Old
2        Middle
3         Young
4         Young
5         Young
6        Middle
7        Middle
8           Old
9        Middle
10        Young
11       Middle
12          Old
13          Old
14       Middle  >
```

In [53]:
```
df.groupby(by = 'Age_category')['Disease'].describe()
```

Out[53]:

	count	unique	top	freq
Age_category				
Middle	7	3	Disease1	3
Old	4	2	Disease1	3
Young	4	3	Disease1	2

In [54]:
```
df.groupby(by = 'Disease')['Age_category'].describe()
```

Out[54]:

Disease	count	unique	top	freq
Disease1	8	3	Middle	3
Disease2	4	3	Middle	2
Disease3	3	2	Middle	2

In [55]:
```
df.loc[(df['Age_category']=='Young') & (df['Disease']=='Disease1')].count()
```

Out[55]:
```
Unnamed: 0      2
Patient ID      2
Patient Name    2
Age             2
Phone           2
Disease         2
Age_category    2
dtype: int64
```

In [ ]:
```

```