In [19]:

```
import pandas as pd
```

In [20]:

```
df=pd.read_csv("health care diabetes.csv")
```

In [21]:

```
df.head()
```

Out[21]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [22]:

```
df.columns
```

Out[22]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [23]:

```
df.corr()
```

Out[23]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

In [24]:

```
df.Insulin.value_counts(normalize=True)
```

Out[24]:

```
0      0.486979
105    0.014323
130    0.011719
140    0.011719
120    0.010417
         ...
73     0.001302
171    0.001302
255    0.001302
52     0.001302
112    0.001302
Name: Insulin, Length: 186, dtype: float64
```

In [25]:

```
df.Insulin.median()
```

Out[25]:

```
30.5
```

In [26]:

```
Insulin_median=df[df['Insulin']!=0]['Insulin'].median()
Insulin_median
```

Out[26]:

```
125.0
```

In [27]:

```
df['Insulin']=df['Insulin'].apply(lambda x: Insulin_median if x==0 else x)
df.head()
```

Out[27]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	125.0	33.6	0.627	50	1
1	1	85	66	29	125.0	26.6	0.351	31	0
2	8	183	64	0	125.0	23.3	0.672	32	1
3	1	89	66	23	94.0	28.1	0.167	21	0
4	0	137	40	35	168.0	43.1	2.288	33	1

In [28]:

```
selected_col=['Glucose','BloodPressure','SkinThickness','BMI']

for i in selected_col:
    median=df[df[i]!=0][i].median()
    df[i]=df[i].apply(lambda x: Insulin_median if x==0 else x)


df.head()
```

Out[28]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	125.0	26.6	0.351	31	0
2	8	183.0	64.0	125.0	125.0	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

In [29]:

```
df.BloodPressure.value_counts(normalize=True).to_frame().iloc[0,:].values[0]
```

Out[29]:

```
0.07421875
```

In [30]:

```
df.Insulin.value_counts(normalize=True).to_frame().iloc[0,:].values[0]*100
```

Out[30]:

```
49.21875
```

In [31]:

```
df.Insulin.value_counts(normalize=True)
```

Out[31]:

```
125.0    0.492188
105.0    0.014323
130.0    0.011719
140.0    0.011719
120.0    0.010417
           ...   
73.0     0.001302
171.0    0.001302
255.0    0.001302
52.0     0.001302
112.0    0.001302
Name: Insulin, Length: 185, dtype: float64
```

In [32]:

```
df.Insulin.median()
```

Out[32]:

```
125.0
```

In [33]:

```
df.dtypes.value_counts()
```

Out[33]:

```
float64    6
int64      3
dtype: int64
```

Week-1

In [34]:

```
df.isnull().any()
```

Out[34]:

```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
```

```
Outcome                     False
dtype: bool
```

In [35]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    float64
 2   BloodPressure             768 non-null    float64
 3   SkinThickness             768 non-null    float64
 4   Insulin                   768 non-null    float64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(6), int64(3)
memory usage: 54.1 KB
```

In [36]:

```
df.columns
```

Out[36]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [37]:

```
df.corr()
```

Out[37]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.127686	0.144984	0.166035	0.025047	0.013372	-0.033523	0.544341	0.221898
Glucose	0.127686	1.000000	0.142632	0.074363	0.418751	0.063797	0.136858	0.266243	0.492985
BloodPressure	0.144984	0.142632	1.000000	0.318976	0.006733	0.317618	-0.039053	0.208816	0.156317
SkinThickness	0.166035	0.074363	0.318976	1.000000	-0.084830	0.117861	-0.130843	0.241883	0.093974
Insulin	0.025047	0.418751	0.006733	-0.084830	1.000000	0.073039	0.126503	0.097101	0.203790
BMI	0.013372	0.063797	0.317618	0.117861	0.073039	1.000000	0.069369	-0.010714	0.129443

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
DiabetesPedigreeFunction	-0.033523	0.136858	-0.039053	-0.130843	0.126503	0.069369	1.000000	0.033561	0.173844
Age	0.544341	0.266243	0.208816	0.241883	0.097101	-0.010714	0.033561	1.000000	0.238356
Outcome	0.221898	0.492985	0.156317	0.093974	0.203790	0.129443	0.173844	0.238356	1.000000

In [38]:

```
data=df
```

In [39]:

```
positive = df[df['Outcome']==1]
positive.head()
```

Out[39]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
2	8	183.0	64.0	125.0	125.0	23.3	0.672	32	1
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1
6	3	78.0	50.0	32.0	88.0	31.0	0.248	26	1
8	2	197.0	70.0	45.0	543.0	30.5	0.158	53	1

In [40]:

```
data['Glucose'].value_counts().head(10)
```

Out[40]:

```
125.0    19
99.0     17
100.0    17
111.0    14
129.0    14
106.0    14
112.0    13
108.0    13
95.0     13
105.0    13
Name: Glucose, dtype: int64
```

In [41]:

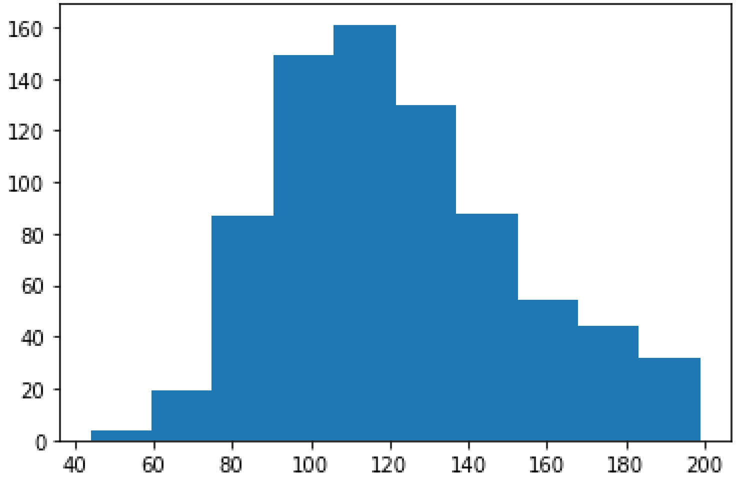
```
import matplotlib.pyplot as plt
```

In [42]:

```
plt.hist(data['Glucose'])
```

Out[42]:

```
(array([  4.,  19.,  87., 149., 161., 130.,  88.,  54.,  44.,  32.]),
 array([ 44. ,  59.5,  75. ,  90.5, 106. , 121.5, 137. , 152.5, 168. ,
        183.5, 199. ]),
 <BarContainer object of 10 artists>)
```



In [43]:

```
data['BloodPressure'].value_counts().head(10)
```

Out[43]:

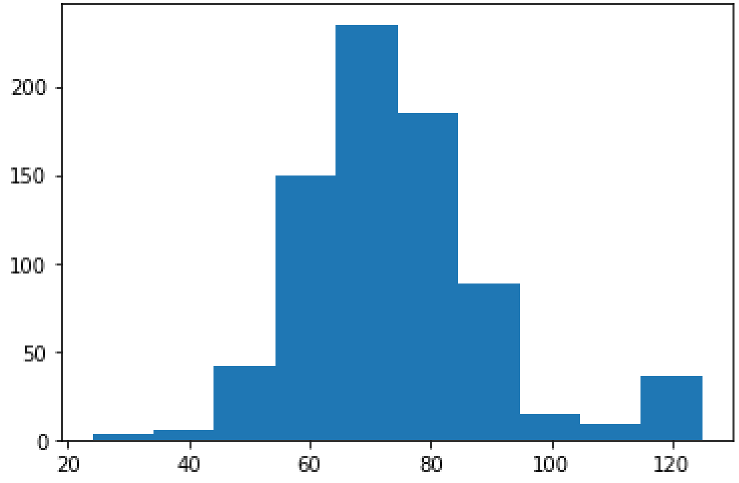
```
70.0     57
74.0     52
78.0     45
68.0     45
72.0     44
64.0     43
80.0     40
76.0     39
60.0     37
125.0    35
Name: BloodPressure, dtype: int64
```

In [44]:

```
plt.hist(data['BloodPressure'])
```

Out[44]:

```
(array([  3.,   6.,  42., 150., 235., 185.,  88.,  14.,   9.,  36.]),
 array([ 24. ,  34.1,  44.2,  54.3,  64.4,  74.5,  84.6,  94.7, 104.8,
        114.9, 125. ]),
 <BarContainer object of 10 artists>)
```



In [45]:

```
data['SkinThickness'].value_counts().head(10)
```

Out[45]:

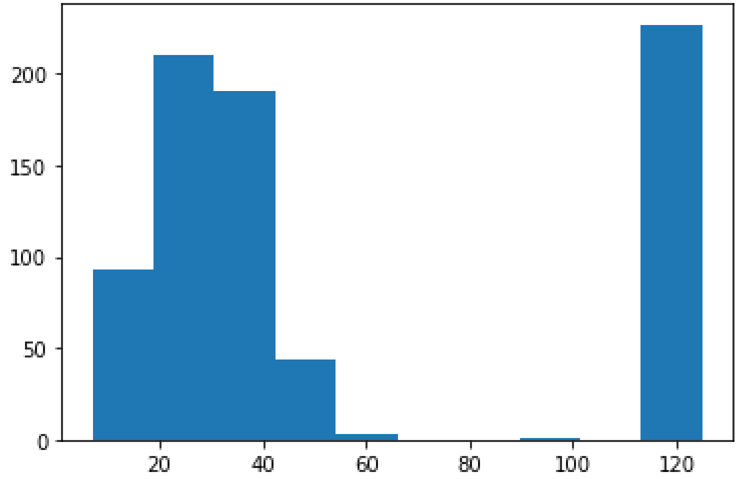
```
125.0    227
32.0      31
30.0      27
27.0      23
23.0      22
33.0      20
28.0      20
18.0      20
31.0      19
19.0      18
Name: SkinThickness, dtype: int64
```

In [46]:

```
plt.hist(data['SkinThickness'])
```

Out[46]:

```
(array([ 93., 210., 190.,  44.,   3.,   0.,   0.,   1.,   0., 227.]),
 array([  7. ,  18.8,  30.6,  42.4,  54.2,  66. ,  77.8,  89.6, 101.4,
        113.2, 125. ]),
 <BarContainer object of 10 artists>)
```



In [47]:

```
data['Insulin'].value_counts().head(10)
```

Out[47]:

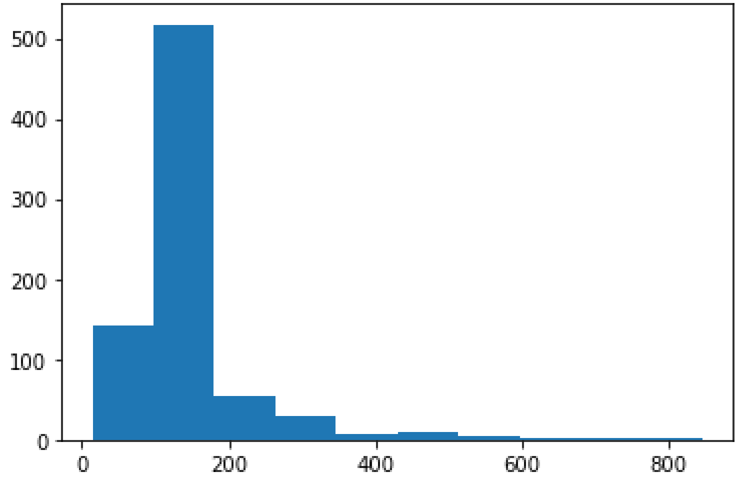
```
125.0    378
105.0     11
130.0      9
140.0      9
120.0      8
94.0       7
180.0      7
100.0      7
135.0      6
115.0      6
Name: Insulin, dtype: int64
```

In [48]:

```
plt.hist(data['Insulin'])
```

Out[48]:

```
(array([142., 517.,  55.,  29.,   7.,  10.,   4.,   1.,   2.,   1.]),
 array([ 14. ,  97.2, 180.4, 263.6, 346.8, 430. , 513.2, 596.4, 679.6,
        762.8, 846. ]),
 <BarContainer object of 10 artists>)
```



In [49]:

```
data['BMI'].value_counts().head(10)
```

Out[49]:

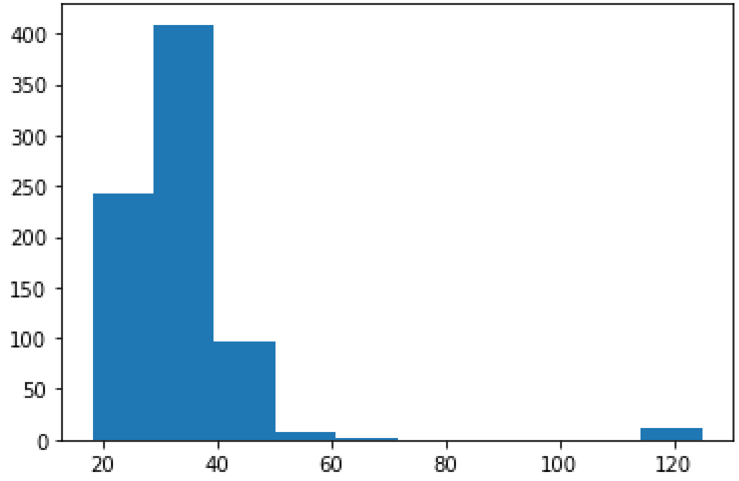
```
32.0     13
31.6     12
31.2     12
125.0    11
32.4     10
33.3     10
30.1      9
32.8      9
32.9      9
30.8      9
Name: BMI, dtype: int64
```

In [50]:

```
plt.hist(data['BMI'])
```

Out[50]:

```
(array([243., 409.,  97.,   7.,   1.,   0.,   0.,   0.,   0.,  11.]),
 array([ 18.2 ,  28.88,  39.56,  50.24,  60.92,  71.6 ,  82.28,  92.96,
        103.64, 114.32, 125.  ]),
 <BarContainer object of 10 artists>)
```



In [51]:

```
data.describe().transpose()
```

Out[51]:

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	121.708333	30.437117	44.000	99.75000	117.0000	140.25000	199.00
BloodPressure	768.0	74.802083	16.333946	24.000	64.00000	73.0000	82.00000	125.00
SkinThickness	768.0	57.483073	44.637491	7.000	25.00000	35.0000	125.00000	125.00
Insulin	768.0	140.671875	86.383060	14.000	121.50000	125.0000	127.25000	846.00
BMI	768.0	33.782943	12.974268	18.200	27.50000	32.4000	36.82500	125.00
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

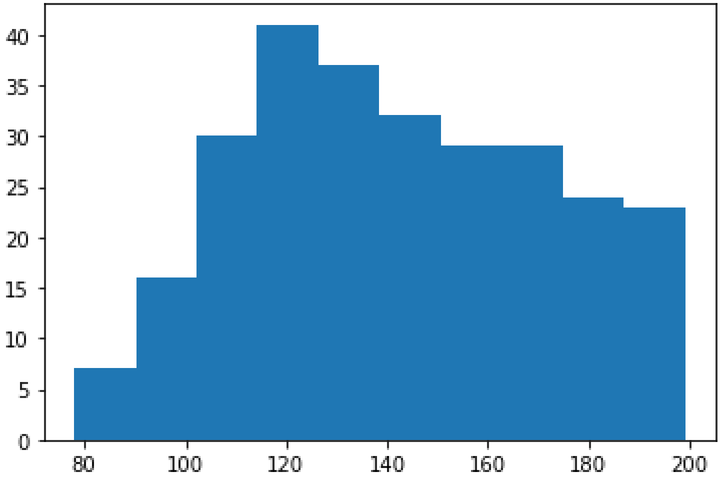
Week-2

In [52]:

```
plt.hist(positive['Glucose'],histtype='stepfilled',bins=10)
```

Out[52]:

```
(array([ 7., 16., 30., 41., 37., 32., 29., 29., 24., 23.]),
 array([ 78. ,  90.1, 102.2, 114.3, 126.4, 138.5, 150.6, 162.7, 174.8,
        186.9, 199. ]),
 [<matplotlib.patches.Polygon at 0x285c148d9d0>])
```



In [53]:

```python
positive['Glucose'].value_counts().head(10)
```

Out[53]:

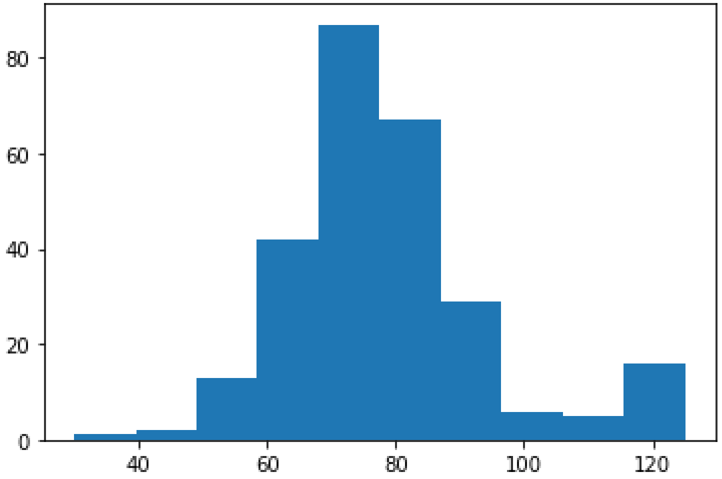
```
125.0    9
128.0    6
129.0    6
115.0    6
158.0    6
146.0    5
124.0    5
162.0    5
173.0    5
109.0    5
Name: Glucose, dtype: int64
```

In [54]:

```python
plt.hist(positive['BloodPressure'],histtype='stepfilled',bins=10)
```

Out[54]:

```
(array([ 1.,  2., 13., 42., 87., 67., 29.,  6.,  5., 16.]),
 array([ 30. ,  39.5,  49. ,  58.5,  68. ,  77.5,  87. ,  96.5, 106. ,
        115.5, 125. ]),
 [<matplotlib.patches.Polygon at 0x285c14fcf40>])
```



In [55]:

```
positive['BloodPressure'].value_counts().head(10)
```

Out[55]:

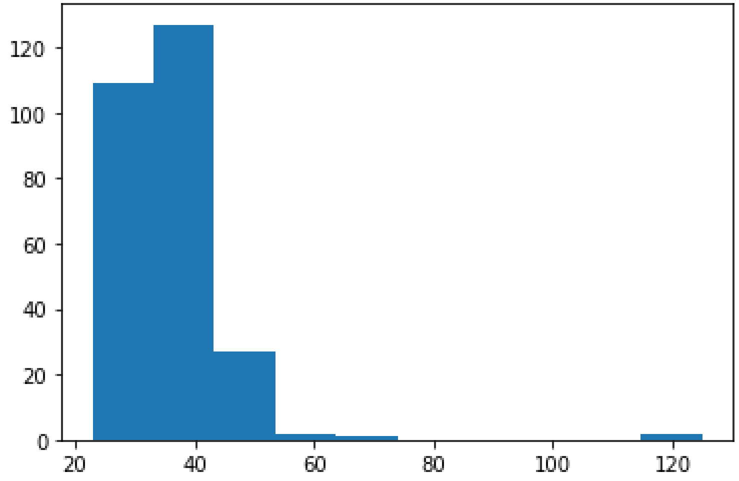
```
70.0     23
76.0     18
78.0     17
74.0     17
72.0     16
125.0    16
80.0     13
64.0     13
82.0     13
84.0     12
Name: BloodPressure, dtype: int64
```

In [56]:

```
plt.hist(positive['BMI'],histtype='stepfilled',bins=10)
```

Out[56]:

```
(array([109., 127.,  27.,   2.,   1.,   0.,   0.,   0.,   0.,   2.]),
 array([ 22.9 ,  33.11,  43.32,  53.53,  63.74,  73.95,  84.16,  94.37,
        104.58, 114.79, 125.  ]),
 [<matplotlib.patches.Polygon at 0x285c15627f0>])
```



In [57]:

```
positive['BMI'].value_counts().head(10)
```

Out[57]:

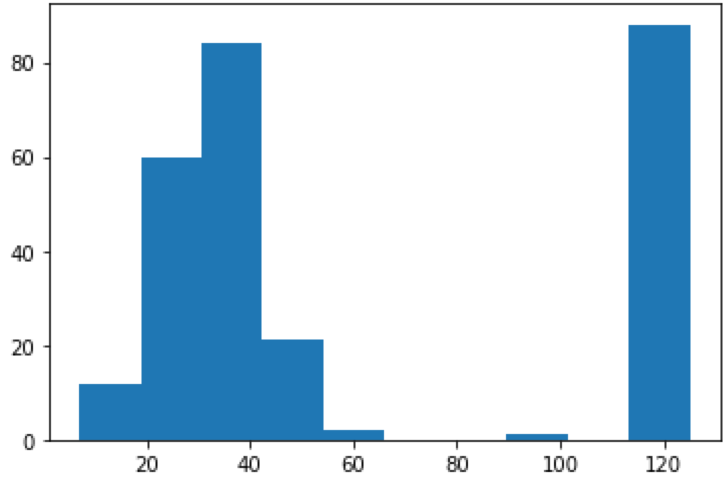
```
32.9    8
31.6    7
33.3    6
31.2    5
30.5    5
32.0    5
34.3    4
30.4    4
32.4    4
43.3    4
Name: BMI, dtype: int64
```

In [58]:

```
plt.hist(positive['SkinThickness'],histtype='stepfilled',bins=10)
```

Out[58]:

```
(array([12., 60., 84., 21.,  2.,  0.,  0.,  1.,  0., 88.]),
 array([  7. ,  18.8,  30.6,  42.4,  54.2,  66. ,  77.8,  89.6, 101.4,
        113.2, 125. ]),
 [<matplotlib.patches.Polygon at 0x285c15ce130>])
```



In [59]:

```
positive['SkinThickness'].value_counts().head(10)
```

Out[59]:

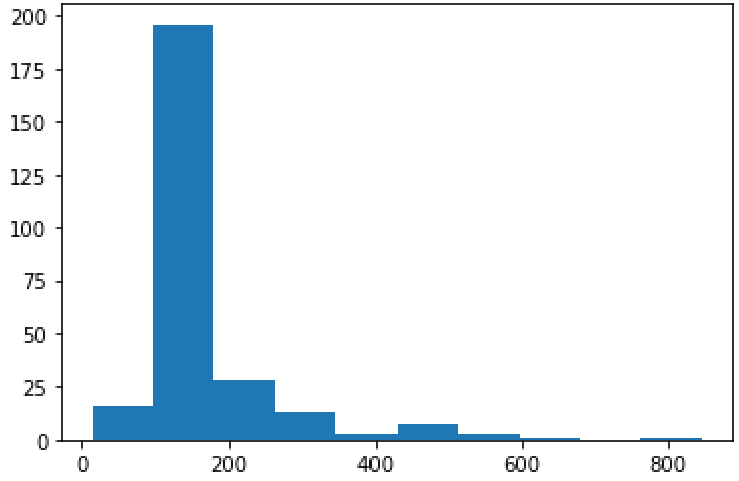
```
125.0    88
32.0     14
30.0      9
33.0      9
39.0      8
37.0      8
36.0      8
35.0      8
27.0      7
29.0      7
Name: SkinThickness, dtype: int64
```

In [60]:

```
plt.hist(positive['Insulin'],histtype='stepfilled',bins=10)
```

Out[60]:

```
(array([ 16., 196.,  28.,  13.,   3.,   7.,   3.,   1.,   0.,   1.]),
 array([ 14. ,  97.2, 180.4, 263.6, 346.8, 430. , 513.2, 596.4, 679.6,
        762.8, 846. ]),
 [<matplotlib.patches.Polygon at 0x285c1623f70>])
```

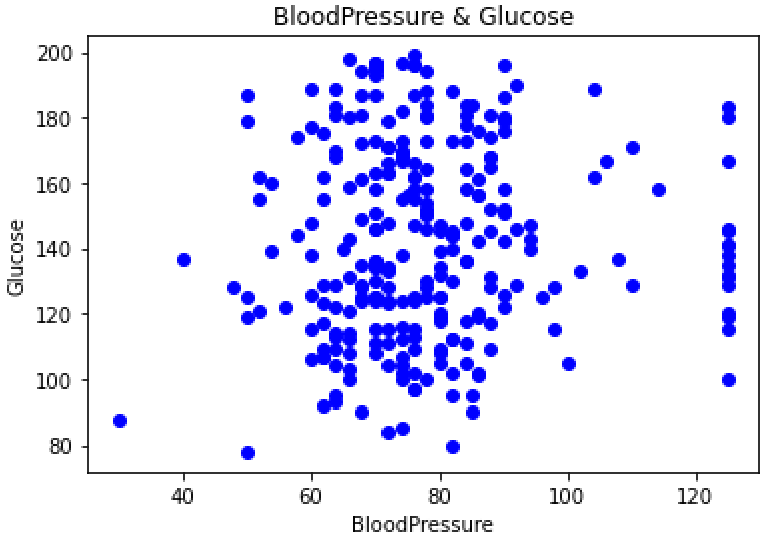


In [61]:
```
positive['Insulin'].value_counts().head(10)
```

Out[61]:
```
125.0    140
130.0      6
180.0      4
175.0      3
156.0      3
185.0      2
225.0      2
155.0      2
114.0      2
160.0      2
Name: Insulin, dtype: int64
```

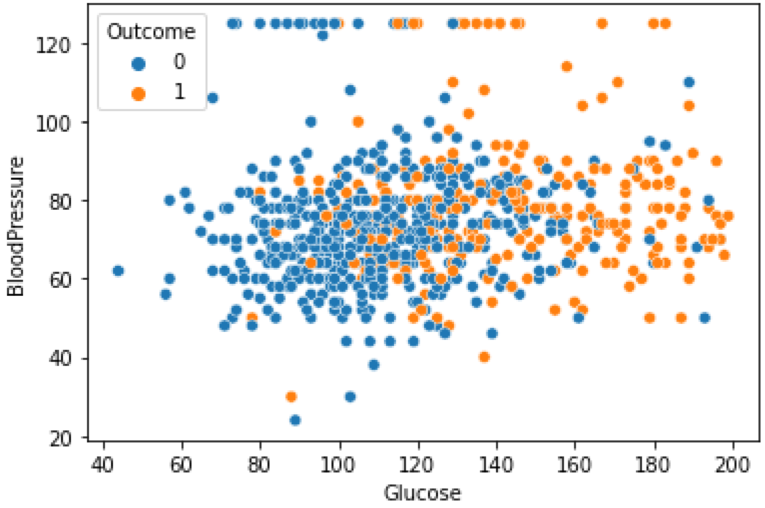
In [62]:
```
BloodPressure = positive['BloodPressure']
Glucose = positive['Glucose']
Insulin = positive['Insulin']
BMI = positive['BMI']
Thickness = positive['SkinThickness']
```

In [63]:
```
plt.scatter(BloodPressure, Glucose, color=['b'])
plt.xlabel('BloodPressure')
plt.ylabel('Glucose')
plt.title('BloodPressure & Glucose')
plt.show()
```



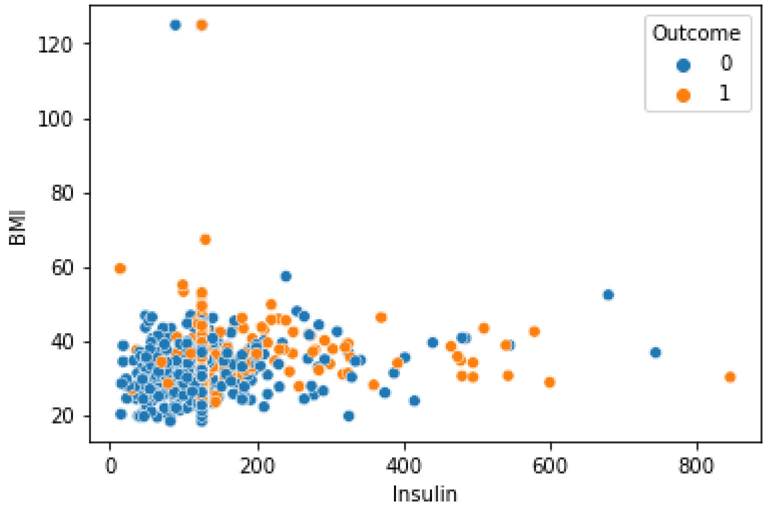
In [64]:
```
import seaborn as sns
```

In [65]:
```
g =sns.scatterplot(x= "Glucose" ,y= "BloodPressure",
                  hue="Outcome",
                  data=data);
```



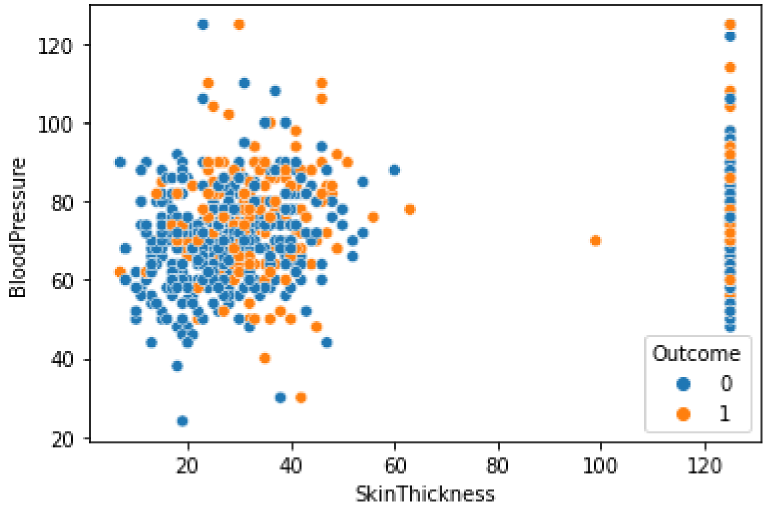
In [66]:

```
g =sns.scatterplot(x= "Insulin" ,y= "BMI",
                   hue="Outcome",
                   data=data);
```



In [67]:

```
g =sns.scatterplot(x= "SkinThickness" ,y= "BloodPressure",
                   hue="Outcome",
                   data=data);
```



In [68]:

```
data.corr()
```

Out[68]:

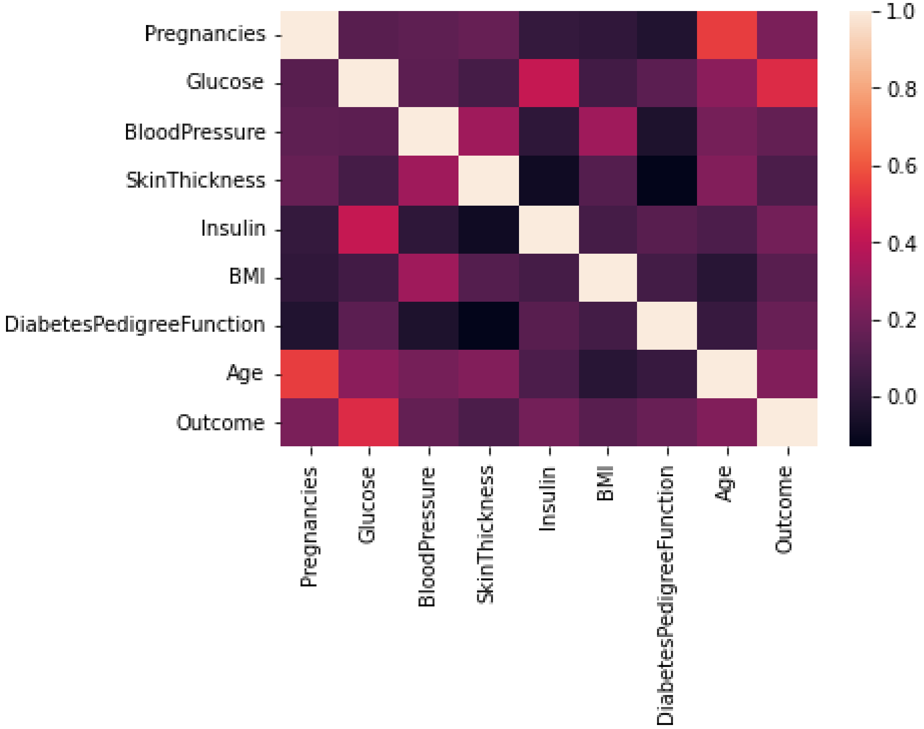
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.127686	0.144984	0.166035	0.025047	0.013372	-0.033523	0.544341	0.221898
Glucose	0.127686	1.000000	0.142632	0.074363	0.418751	0.063797	0.136858	0.266243	0.492985
BloodPressure	0.144984	0.142632	1.000000	0.318976	0.006733	0.317618	-0.039053	0.208816	0.156317
SkinThickness	0.166035	0.074363	0.318976	1.000000	-0.084830	0.117861	-0.130843	0.241883	0.093974
Insulin	0.025047	0.418751	0.006733	-0.084830	1.000000	0.073039	0.126503	0.097101	0.203790
BMI	0.013372	0.063797	0.317618	0.117861	0.073039	1.000000	0.069369	-0.010714	0.129443
DiabetesPedigreeFunction	-0.033523	0.136858	-0.039053	-0.130843	0.126503	0.069369	1.000000	0.033561	0.173844
Age	0.544341	0.266243	0.208816	0.241883	0.097101	-0.010714	0.033561	1.000000	0.238356
Outcome	0.221898	0.492985	0.156317	0.093974	0.203790	0.129443	0.173844	0.238356	1.000000

In [69]:

```
sns.heatmap(data.corr())
```

Out[69]:

```
<AxesSubplot:>
```

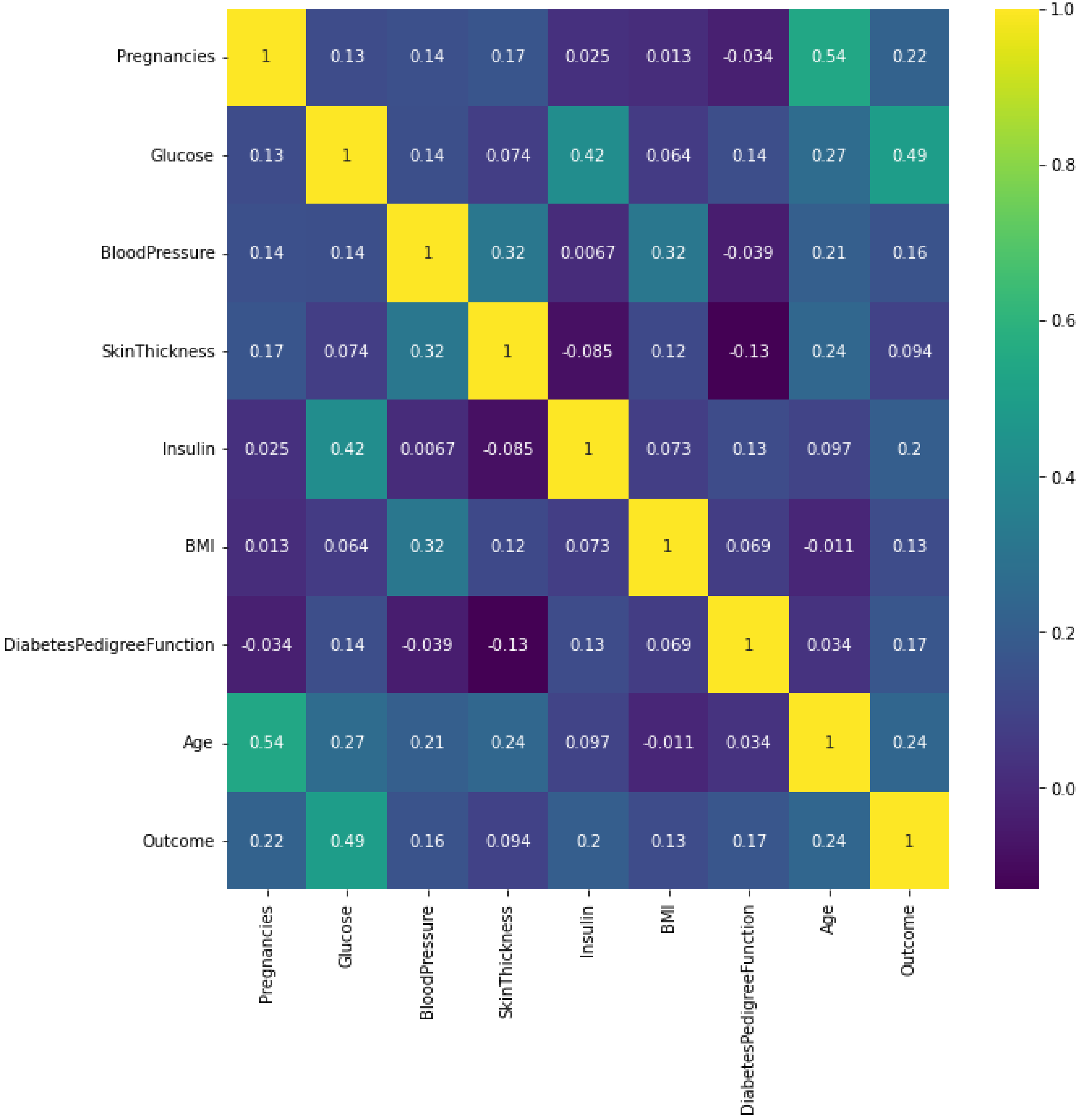


In [70]:

```
plt.subplots(figsize=(10,10))
sns.heatmap(data.corr(),annot=True,cmap='viridis')
```

Out[70]:

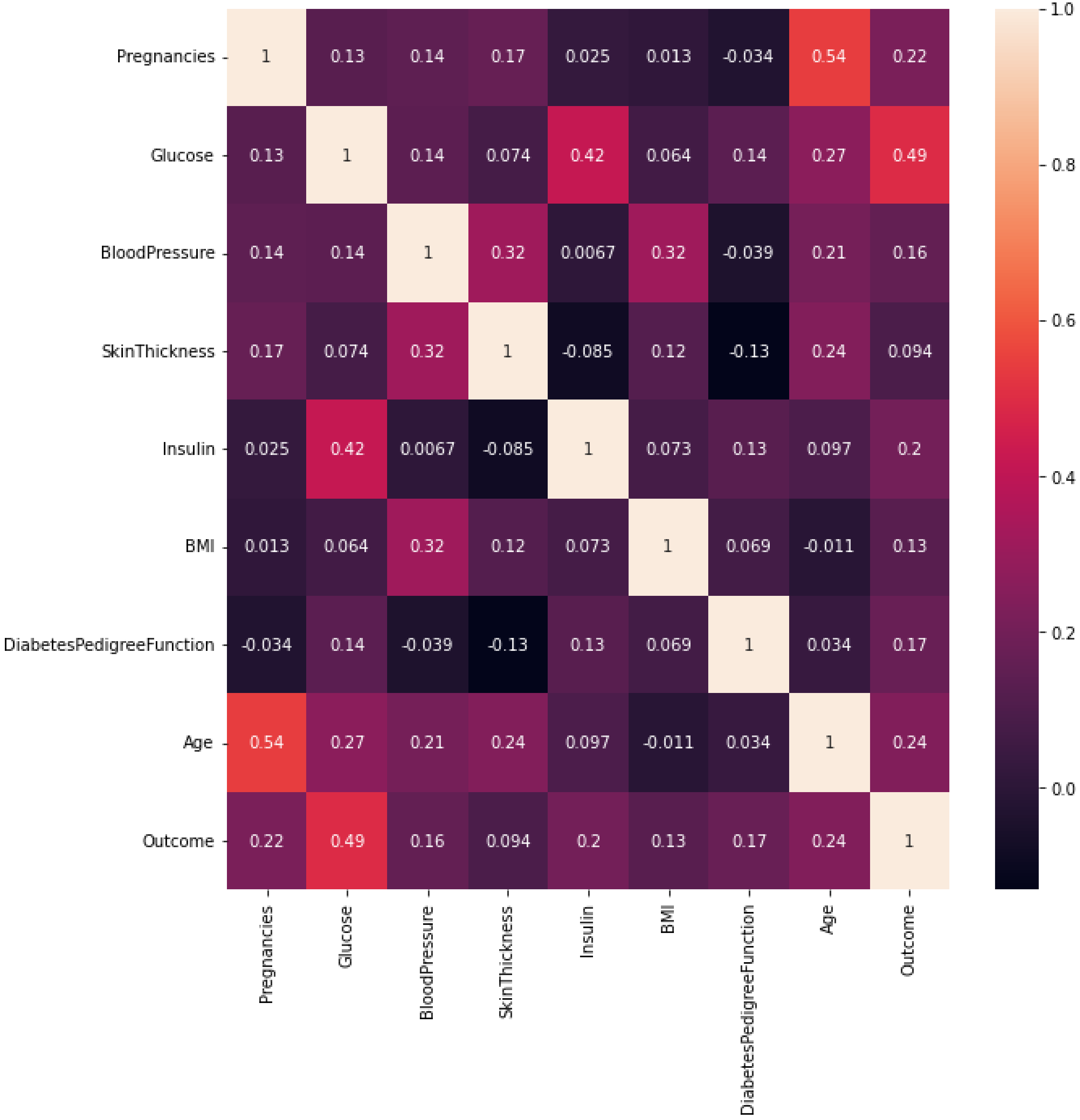
```
<AxesSubplot:>
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.14	0.17	0.025	0.013	-0.034	0.54	0.22
Glucose	0.13	1	0.14	0.074	0.42	0.064	0.14	0.27	0.49
BloodPressure	0.14	0.14	1	0.32	0.0067	0.32	-0.039	0.21	0.16
SkinThickness	0.17	0.074	0.32	1	-0.085	0.12	-0.13	0.24	0.094
Insulin	0.025	0.42	0.0067	-0.085	1	0.073	0.13	0.097	0.2
BMI	0.013	0.064	0.32	0.12	0.073	1	0.069	-0.011	0.13
DiabetesPedigreeFunction	-0.034	0.14	-0.039	-0.13	0.13	0.069	1	0.034	0.17
Age	0.54	0.27	0.21	0.24	0.097	-0.011	0.034	1	0.24
Outcome	0.22	0.49	0.16	0.094	0.2	0.13	0.17	0.24	1

```
In [71]: plt.subplots(figsize=(10,10))
         sns.heatmap(data.corr(),annot=True)
```

```
Out[71]: <AxesSubplot:>
```


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1 0.13 0.14 0.17 0.025 0.013 -0.034 0.54 0.22
0.13 1 0.14 0.074 0.42 0.064 0.14 0.27 0.49
0.14 0.14 1 0.32 0.0067 0.32 -0.039 0.21 0.16
0.17 0.074 0.32 1 -0.085 0.12 -0.13 0.24 0.094
0.025 0.42 0.0067 -0.085 1 0.073 0.13 0.097 0.2
0.013 0.064 0.32 0.12 0.073 1 0.069 -0.011 0.13
-0.034 0.14 -0.039 -0.13 0.13 0.069 1 0.034 0.17
0.54 0.27 0.21 0.24 0.097 -0.011 0.034 1 0.24
0.22 0.49 0.16 0.094 0.2 0.13 0.17 0.24 1
1.0
0.8
0.6
0.4
0.2
0.0

Week-3

In [72]:

```
data.head()
```

Out[72]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.0	125.0	33.6	0.627	50	1
1	1	85.0	66.0	29.0	125.0	26.6	0.351	31	0
2	8	183.0	64.0	125.0	125.0	23.3	0.672	32	1
3	1	89.0	66.0	23.0	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35.0	168.0	43.1	2.288	33	1

In [73]:

```
features = data.iloc[:,[0,1,2,3,4,6,7]].values
label = data.iloc[:,8].values
```

In [74]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(features,label,test_size=0.2,random_state =10)
```

In [94]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train,y_train)
import warnings
warnings.filterwarnings('ignore')
```

In [95]:

```
model
```

Out[95]:

```
LogisticRegression()
```

In [76]:

```
print(model.score(X_train,y_train))
print(model.score(X_test,y_test))
```

```
0.7671009771986971
```

```
0.7142857142857143
```

In [77]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(label,model.predict(features))
cm
```

Out[77]:

```
array([[443,  57],
       [130, 138]], dtype=int64)
```

In [78]:

```
from sklearn.metrics import classification_report
print(classification_report(label,model.predict(features)))
```

```
              precision    recall  f1-score   support

           0       0.77      0.89      0.83       500
           1       0.71      0.51      0.60       268

    accuracy                           0.76       768
   macro avg       0.74      0.70      0.71       768
weighted avg       0.75      0.76      0.75       768
```

Week-4

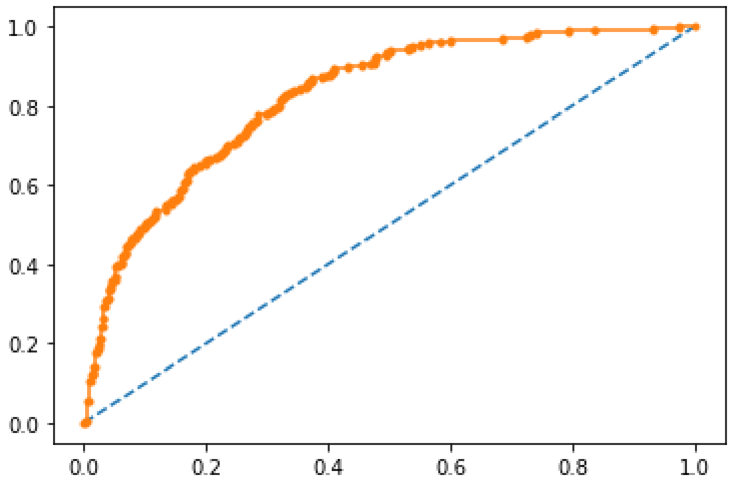
In [79]:

```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

probs = model.predict_proba(features)

probs = probs[:, 1]

auc = roc_auc_score(label, probs)
print('AUC: %.3f' %auc)

fpr, tpr, threshold = roc_curve(label, probs)

plt.plot([0, 1], [0, 1], linestyle='--')

plt.plot(fpr, tpr, marker='.')
```

```
AUC: 0.825
```

Out[79]:

```
[<matplotlib.lines.Line2D at 0x285c2d877c0>]
```



In [80]:

```python
from sklearn.tree import DecisionTreeClassifier
model3 = DecisionTreeClassifier(max_depth=5)
model3.fit(X_train,y_train)
```

Out[80]:

```
DecisionTreeClassifier(max_depth=5)
```

In [81]:

```python
model3.score(X_train,y_train)
```

Out[81]:

```
0.8175895765472313
```

In [82]:

```python
model3.score(X_test,y_test)
```

Out[82]:

```
0.7272727272727273
```

In [83]:

```python
from sklearn.ensemble import RandomForestClassifier
model4 = RandomForestClassifier(n_estimators=11)
model4.fit(X_train,y_train)
```

Out[83]:

```
RandomForestClassifier(n_estimators=11)
```

In [84]:

```
model4.score(X_train,y_train)
```

Out[84]:

```
0.990228013029316
```

In [85]:

```
model4.score(X_test,y_test)
```

Out[85]:

```
0.6948051948051948
```

In [86]:

```
from sklearn.svm import SVC
model5 = SVC(kernel='rbf',gamma='auto')
model5.fit(X_train,y_train)
```

Out[86]:

```
SVC(gamma='auto')
```

In [87]:

```
model5.score(X_test,y_test,)
```

Out[87]:

```
0.6168831168831169
```

In [88]:

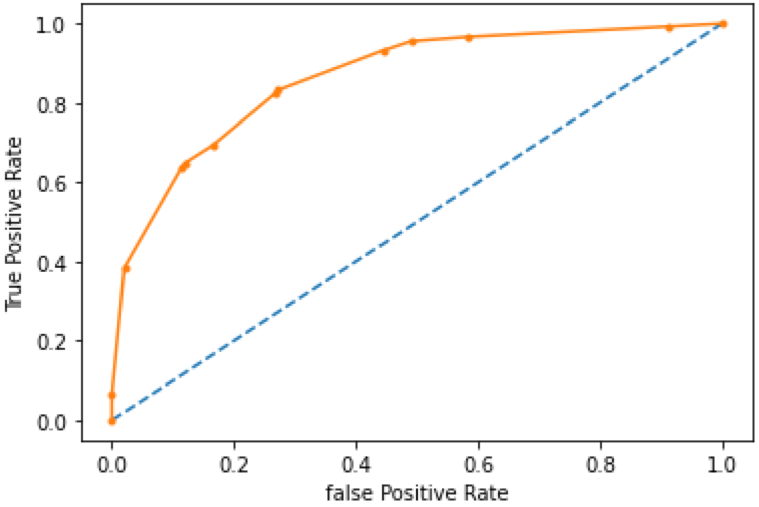
```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

probs = model3.predict_proba(features)

probs = probs[:, 1]

auc = roc_auc_score(label, probs)
print('AUC: %.3f' %auc)

fpr, tpr, thresholds = roc_curve(label, probs)
print("true Positive Rate - {}, False Positive Rate - {} Threshold - {}".format(tpr,fpr,thresholds))

plt.plot([0, 1], [0, 1], linestyle='--')

plt.plot(fpr, tpr, marker='.')
plt.xlabel("false Positive Rate")
plt.ylabel("True Positive Rate")
```

```
AUC: 0.862
```

```
true Positive Rate - [0.         0.06343284 0.38432836 0.6380597  0.64925373 0.69402985
 0.82462687 0.83208955 0.93283582 0.95522388 0.96641791 0.99253731
 1.        ], False Positive Rate - [0.    0.    0.02  0.114 0.12  0.166 0.268 0.272 0.446 0.49  0.584 0.912
 1.   ] Threshold - [2.         1.         0.91139241 0.6        0.5        0.4
 0.39130435 0.33333333 0.20652174 0.18181818 0.05       0.03521127
 0.        ]
```

Out[88]:

```
Text(0, 0.5, 'True Positive Rate')
```



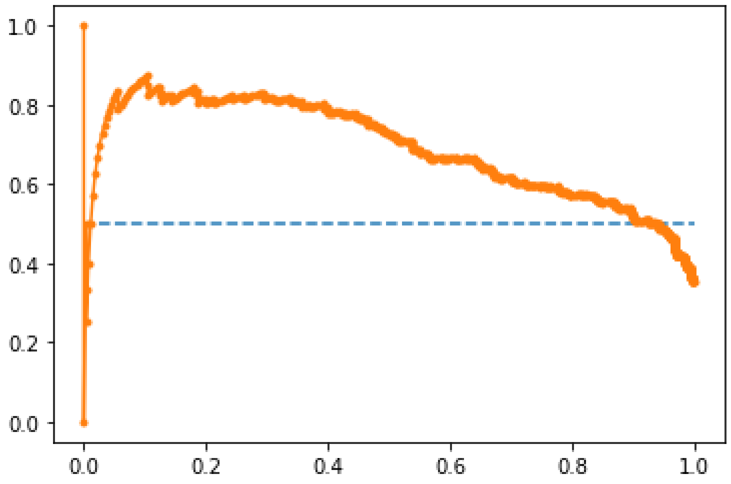
In [89]:

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
probs = model.predict_proba(features)
probs = probs[:, 1]

yhat = model.predict(features)

precision, recall, thresholds = precision_recall_curve(label, probs)

f1 = f1_score(label, yhat)

auc = auc(recall, precision)

ap = average_precision_score(label, probs)
print('f1=%.3f auc=%.3f ap=%.3f' %(f1, auc, ap))
print('f1=%.3f auc=%.3f ap=%.3f' %(f1, auc, ap))
plt.plot([0, 1], [0.5, 0.5], linestyle='--')

plt.plot(recall, precision, marker='.')
```

```
f1=0.596 auc=0.689 ap=0.692
f1=0.596 auc=0.689 ap=0.692
```

Out[89]:

```
[<matplotlib.lines.Line2D at 0x285c310c910>]
```



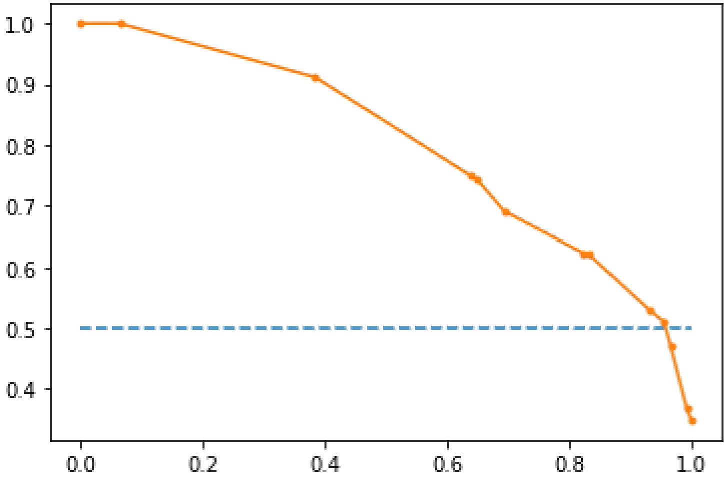
In [90]:

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
probs = model3.predict_proba(features)
probs = probs[:, 1]

yhat = model.predict(features)

precision, recall, thresholds = precision_recall_curve(label, probs)

f1 = f1_score(label, yhat)

auc = auc(recall, precision)

print('f1=%.3f auc=%.3f ap=%.3f' %(f1, auc, ap))

plt.plot([0, 1], [0.5, 0.5], linestyle='--')
plt.plot(recall, precision, marker='.')
```

```
f1=0.596 auc=0.801 ap=0.692
```

Out[90]:

```
[<matplotlib.lines.Line2D at 0x285c2966880>]
```



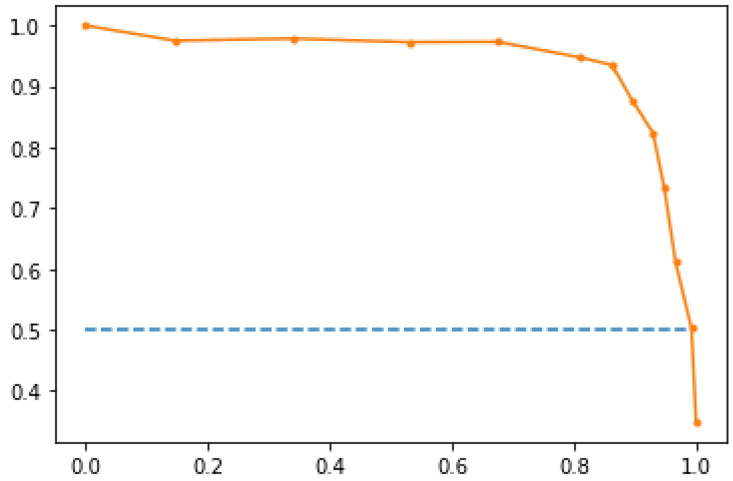
In [91]:

```
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
probs = model4.predict_proba(features)
probs = probs[:, 1]

yhat = model.predict(features)

precision, recall, thresholds = precision_recall_curve(label, probs)

f1 = f1_score(label, yhat)

auc = auc(recall, precision)

print('f1=%.3f auc=%.3f ap=%.3f' %(f1, auc, ap))

plt.plot([0, 1], [0.5, 0.5], linestyle='--')

plt.plot(recall, precision, marker='.')
```

```
f1=0.596 auc=0.942 ap=0.692
```

Out[91]:

```
[<matplotlib.lines.Line2D at 0x285c2936e80>]
```



Project submited by :- Sambit Mahanta

In [ ]:

In [ ]: