Data Analysis and Visualization

The Pima are a group of native Americans living in what is now central and southern Arizona. The Pima Indians of Arizona have the highest rate of obesity and diabetes ever recorded, and since they have the willingness to help the research process, the National Institute of Diabetes and Digestive and Kidney have been able to collect the data about the Pima's group (only women are included in this study).

Columns

pregnant: It represents the number of times the woman got pregnant during her life.

glucose: It represents the plasma glucose concentration at 2 hours in an oral glucose tolerance test.

diastolic: The blood pressure is a very well-known way to measure the health of the heart of a person, there are too measure in fact, the diastolic and the systolic. In this data set, we have the diastolic which is in the fact the pressure in (mm/Hg) when the heart relaxed after the contraction.

triceps: It is a value used to estimate body fat (mm) which is measured on the right arm halfway between the olecranon process of the elbow and the acromial process of the scapula.

insulin: It represents the rate of insulin 2 hours serum insulin (mu U/ml).

bmi: It represents the Body Mass Index (weight in kg / (height in meters squared), and is an indicator of the health of a person.

diabetes: It is an indicator of history of diabetes in the family.

age: It represents the age in years of the Pima's woman.

test: It can take only 2 values (‘negatif’ or ‘positif’) and represents if the patient shows signs of diabetes.

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
df=pd.read_csv('PimaIndians.csv')
```

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

Out[6]:

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
0	1	89	66	23	94	28.1	0.167	21	negatif
1	0	137	40	35	168	43.1	2.288	33	positif
2	3	78	50	32	88	31.0	0.248	26	positif
3	2	197	70	45	543	30.5	0.158	53	positif
4	1	189	60	23	846	30.1	0.398	59	positif

In [7]:
```
df['test'] = df['test'].map({'positif': 1, 'negatif': 0})
```

In [8]:
```
df.head()
```

Out[8]:

	pregnant	glucose	diastolic	triceps	insulin	bmi	diabetes	age	test
0	1	89	66	23	94	28.1	0.167	21	0
1	0	137	40	35	168	43.1	2.288	33	1
2	3	78	50	32	88	31.0	0.248	26	1
3	2	197	70	45	543	30.5	0.158	53	1
4	1	189	60	23	846	30.1	0.398	59	1

In [9]:
```
df['test'].value_counts()
```

Out[9]:
```
0    262
1    130
Name: test, dtype: int64
```

In [10]:
```
df.columns
```

Out[10]:
```
Index([u'pregnant', u'glucose', u'diastolic', u'triceps', u'insulin',
u'bmi',
       u'diabetes', u'age', u'test'],
      dtype='object')
```

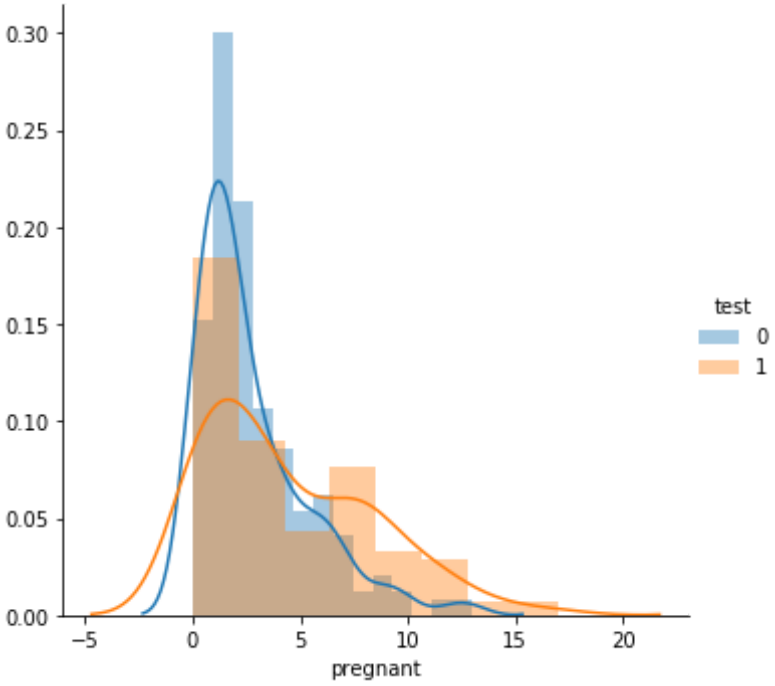
In [11]:
```
df.shape
```

Out[11]:
```
(392, 9)
```

Data Visualization

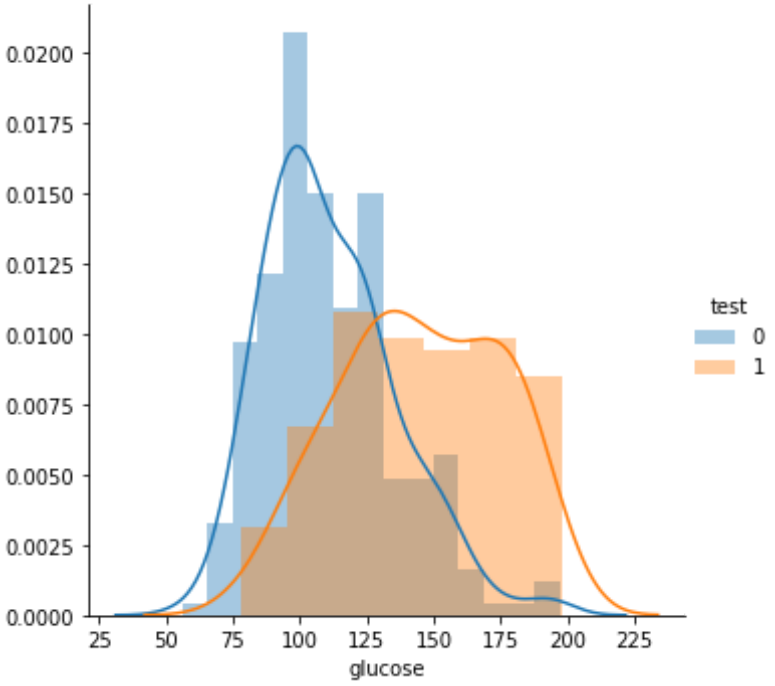
Probability Density Functions

In [12]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'pregnant')\
.add_legend()
plt.show()
```



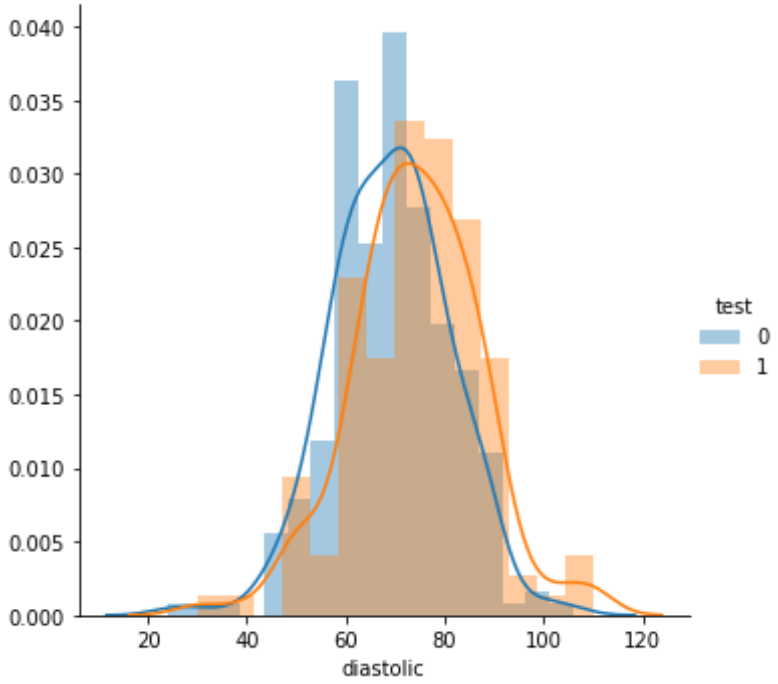
Here Classes 0 and 1 have overlapped pdfs.

In [13]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'glucose')\
.add_legend()
plt.show()
```



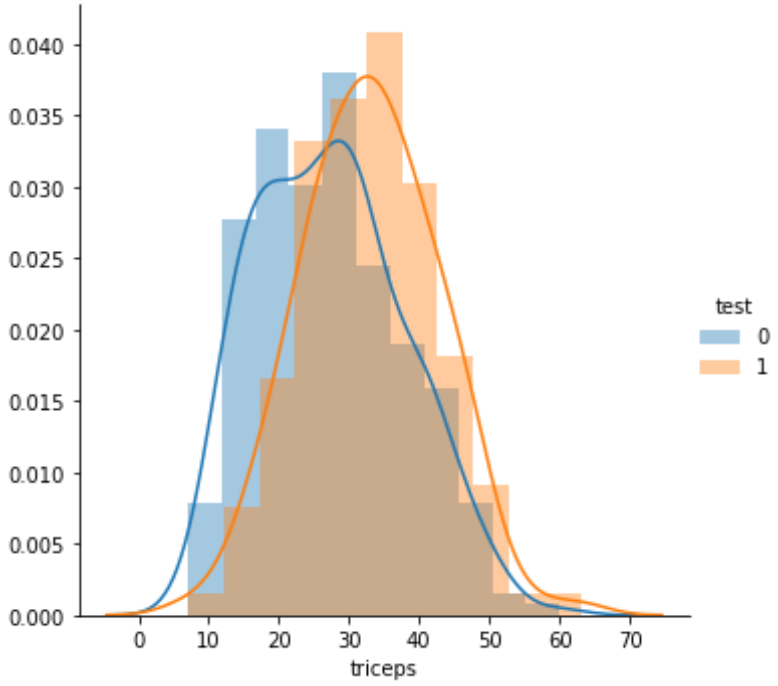
Here Classes 0 and 1 don't have overlapped pdfs.

In [14]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'diastolic')\
.add_legend()
plt.show()
```



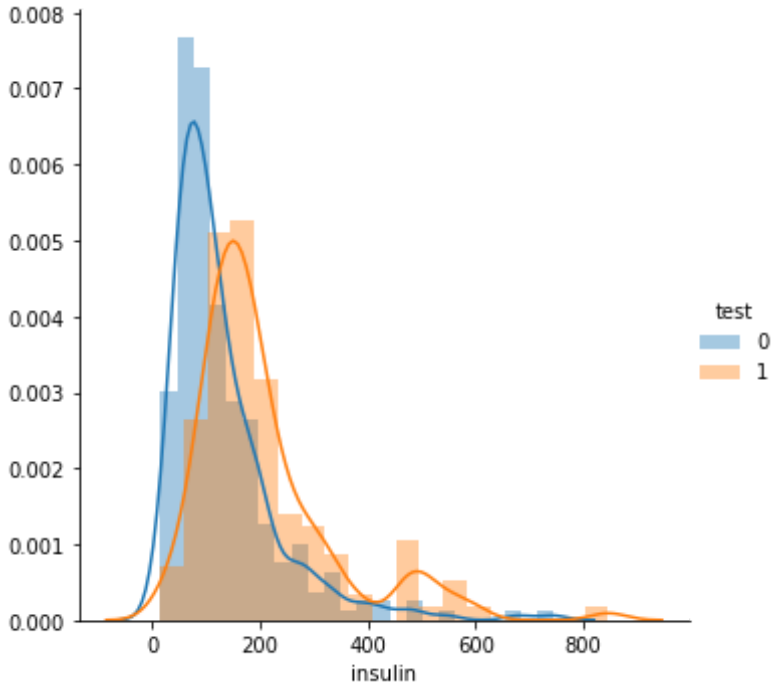
Here Classes 0 and 1 have almost overlapped pdfs.

In [15]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'triceps')\
.add_legend()
plt.show()
```



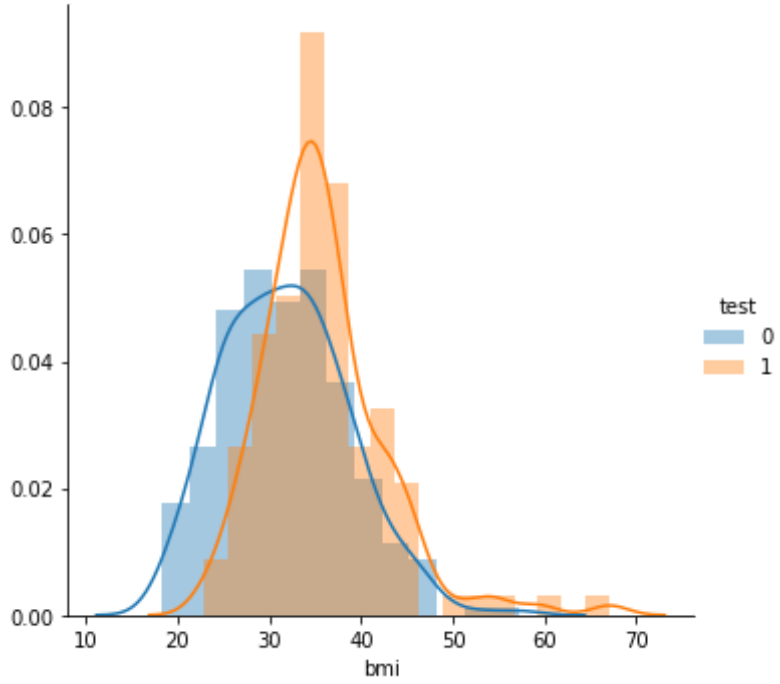
Here Classes 0 and 1 have almost overlapped pdfs.

In [16]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'insulin')\
.add_legend()
plt.show()
```



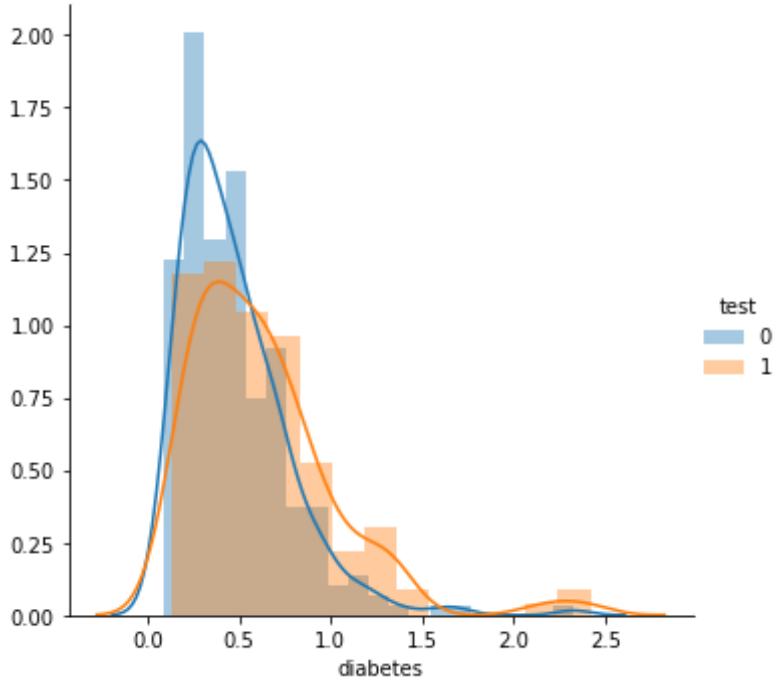
Here Classes 0 and 1 don't have overlapped pdfs.

In [17]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'bmi')\
.add_legend()
plt.show()
```



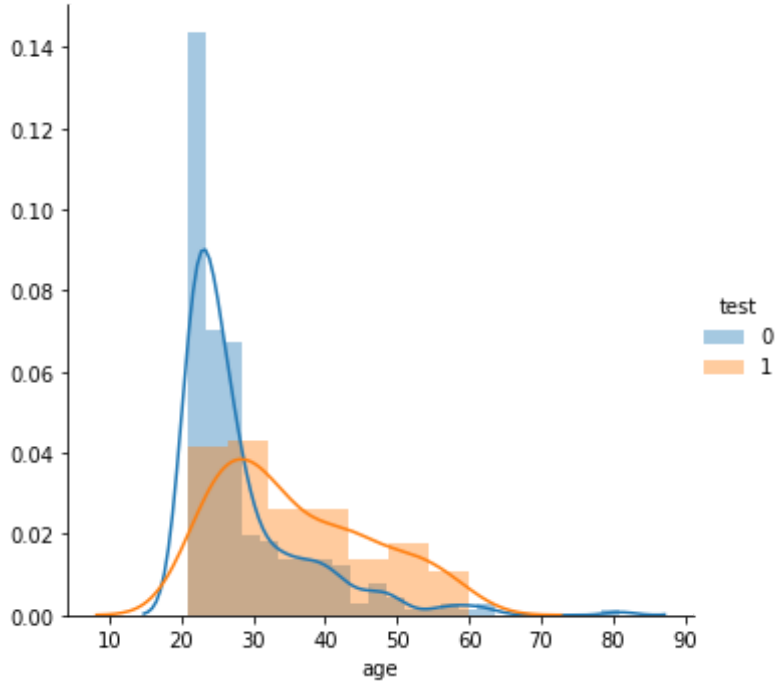
Here Classes 0 and 1 have almost overlapped pdfs.

In [18]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'diabetes')\
.add_legend()
plt.show()
```



Here Classes 0 and 1 have almost overlapped pdfs.

In [19]:
```
sns.FacetGrid(df,hue="test",height=5)\
.map(sns.distplot,'age')\
.add_legend()
plt.show()
```



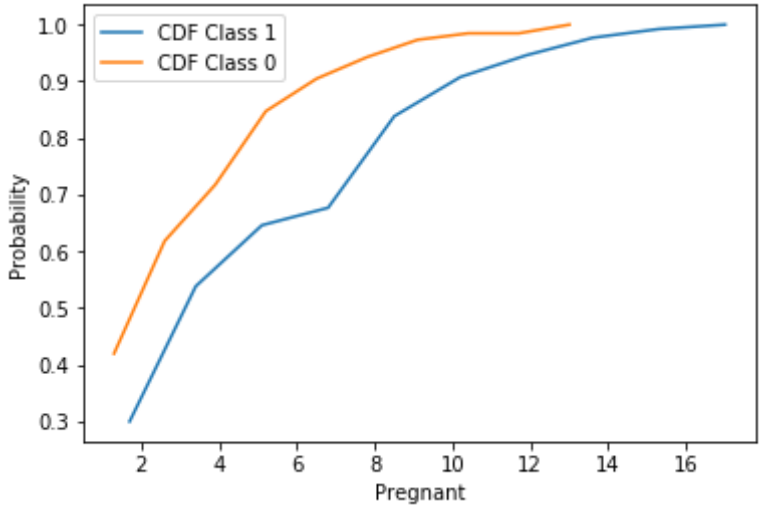
Here Classes 0 and 1 have almost overlapped pdfs.

Cumulative Density Functions

In [20]:
```
one=df.loc[df['test']==1]
zero=df.loc[df['test']==0]
```

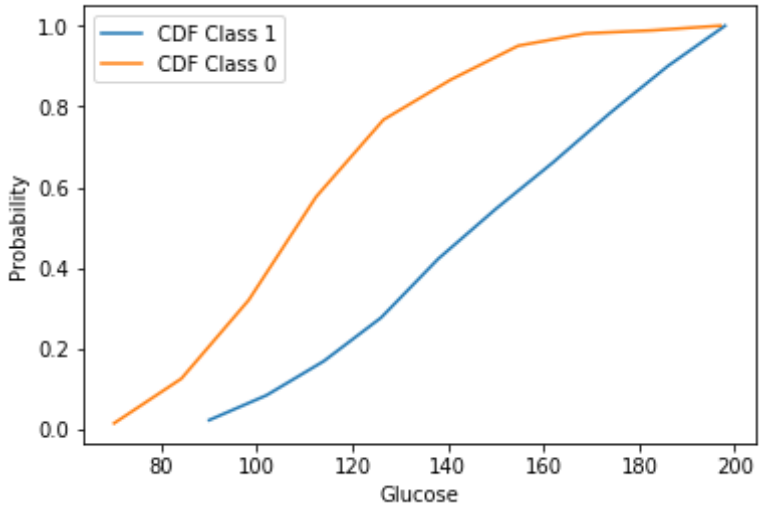
In [21]:
```
counts,bin_edges=np.histogram(one['pregnant'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['pregnant'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
```

```
plt.xlabel("Pregnant")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



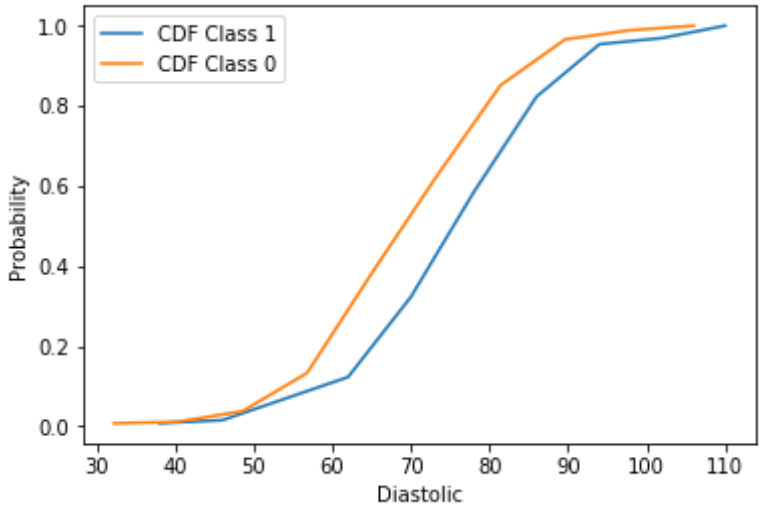
No such difference in CDFs

In [22]:
```
counts,bin_edges=np.histogram(one['glucose'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['glucose'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Glucose")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



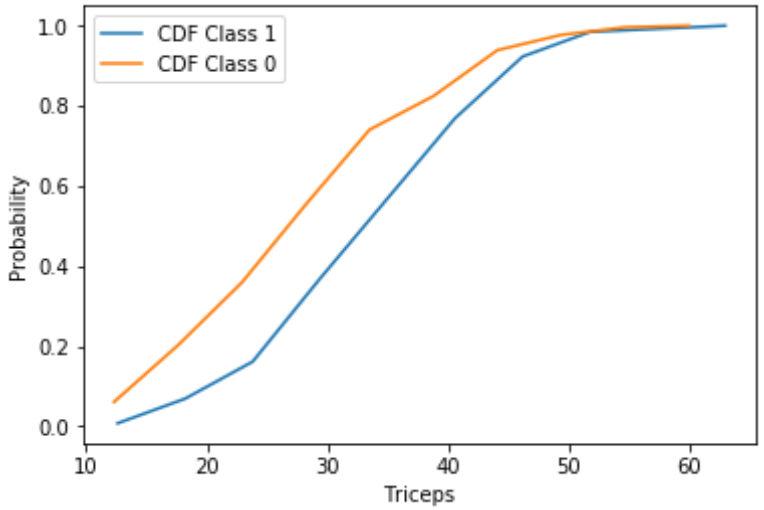
Here at about 120 Glucose level we can make a classifier as there is significant difference in the probability. For class 1, People with Glucose level less than 120 are about 20% while for class 2, people with glucose level less than 120 are about 60%.

In [23]:
```
counts,bin_edges=np.histogram(one['diastolic'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['diastolic'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Diastolic")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



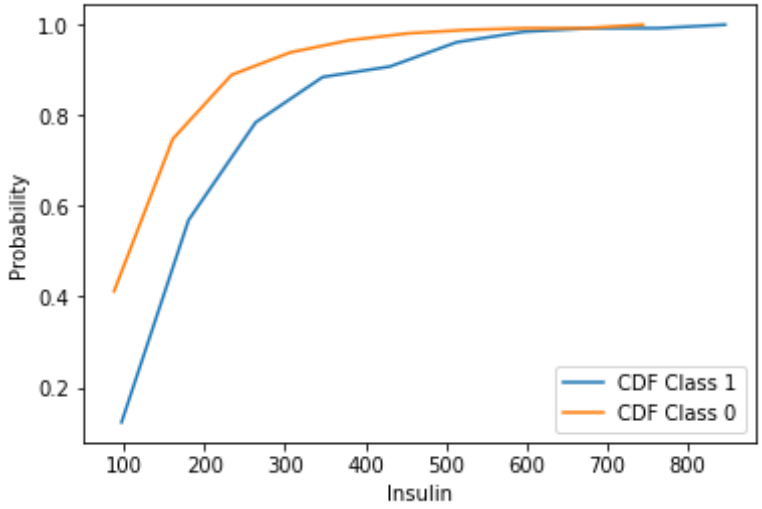
No such difference in CDFs

In [24]:
```
counts,bin_edges=np.histogram(one['triceps'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['triceps'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Triceps")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



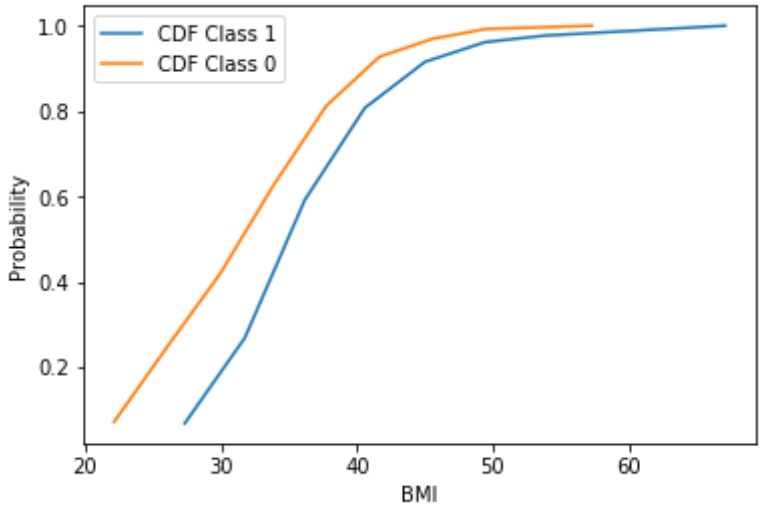
No such difference in CDFs

In [25]:
```
counts,bin_edges=np.histogram(one['insulin'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['insulin'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Insulin")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



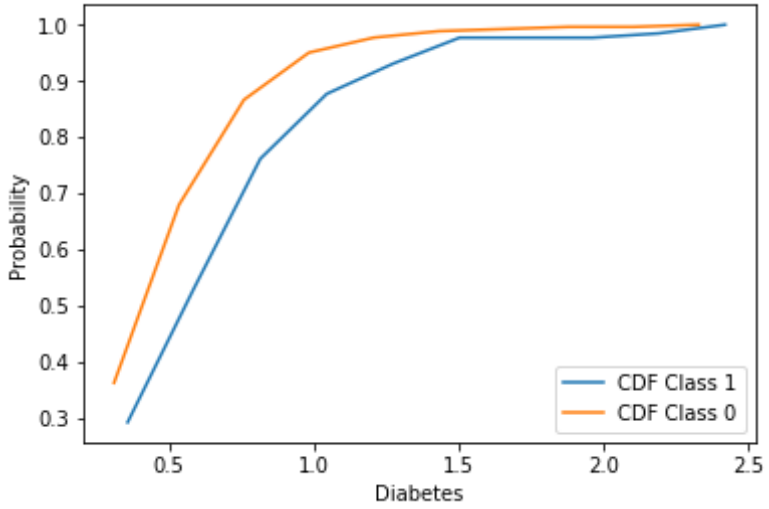
Here at about 150 insulin level we can make a classifier as there is significant difference in the probability. For class 1, People with insulin level less than 110 are about 30% while for class 2, people with insulin level less than 110 are about 60%.

In [26]:
```
counts,bin_edges=np.histogram(one['bmi'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['bmi'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("BMI")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



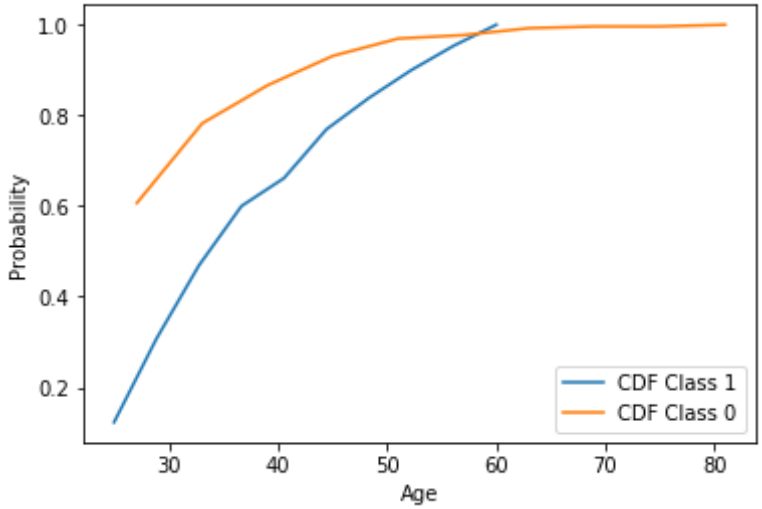
No such difference in CDFs

In [27]:
```
counts,bin_edges=np.histogram(one['diabetes'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['diabetes'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Diabetes")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



No such difference in CDFs

In [28]:
```
counts,bin_edges=np.histogram(one['age'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(zero['age'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 0")
plt.xlabel("Age")
plt.ylabel("Probability")
plt.legend()
plt.show()
```



In [ ]:

Here at about at age 25 we can make a classifier as there is significant difference in the probability. For class 1, People with age less than 25 are about 10% while for class 2, people with age less than 25 are about 60%.

Conclusion

So, here we can use features: Glucose, Insulin, Age to make the classifier.