In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix, accuracy_score,r2_score, classification_report
# from sklearn.externals.six import StringIO
from IPython.display import Image
```

In [2]:
```python
dataset = pd.read_csv("D:/Live_in_lab/heartdata.csv")
```

In [3]:
```python
dataset.head(10)
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	num
0	28	1	2	130.0	132.000000	0.0	2.0	185.0	0.0	0.0	0
1	29	1	2	120.0	243.000000	0.0	0.0	160.0	0.0	0.0	0
2	29	1	2	140.0	250.848708	0.0	0.0	170.0	0.0	0.0	0
3	30	0	1	170.0	237.000000	0.0	1.0	170.0	0.0	0.0	0
4	31	0	2	100.0	219.000000	0.0	1.0	150.0	0.0	0.0	0
5	32	0	2	105.0	198.000000	0.0	0.0	165.0	0.0	0.0	0
6	32	1	2	110.0	225.000000	0.0	0.0	184.0	0.0	0.0	0
7	32	1	2	125.0	254.000000	0.0	0.0	155.0	0.0	0.0	0
8	33	1	3	120.0	298.000000	0.0	0.0	185.0	0.0	0.0	0
9	34	0	2	130.0	161.000000	0.0	0.0	190.0	0.0	0.0	0

In [5]:
```python
dataset.shape
df=dataset[num]
sns.pairplot(df,hue="heart disease")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [5], in <cell line: 2>()
      1 dataset.shape
----> 2 df=dataset[num]
      3 sns.pairplot(df,hue="heart disease")

NameError: name 'num' is not defined
```

In [6]:
```python
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 293 entries, 0 to 292
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       293 non-null    int64
 1   sex       293 non-null    int64
 2   cp        293 non-null    int64
 3   trestbps  293 non-null    float64
 4   chol      293 non-null    float64
 5   fbs       293 non-null    float64
 6   restecg   293 non-null    float64
 7   thalach   293 non-null    float64
 8   exang     293 non-null    float64
 9   oldpeak   293 non-null    float64
 10  num       293 non-null    int64
dtypes: float64(7), int64(4)
memory usage: 25.3 KB
```

In [7]:
```python
dataset.describe()
```

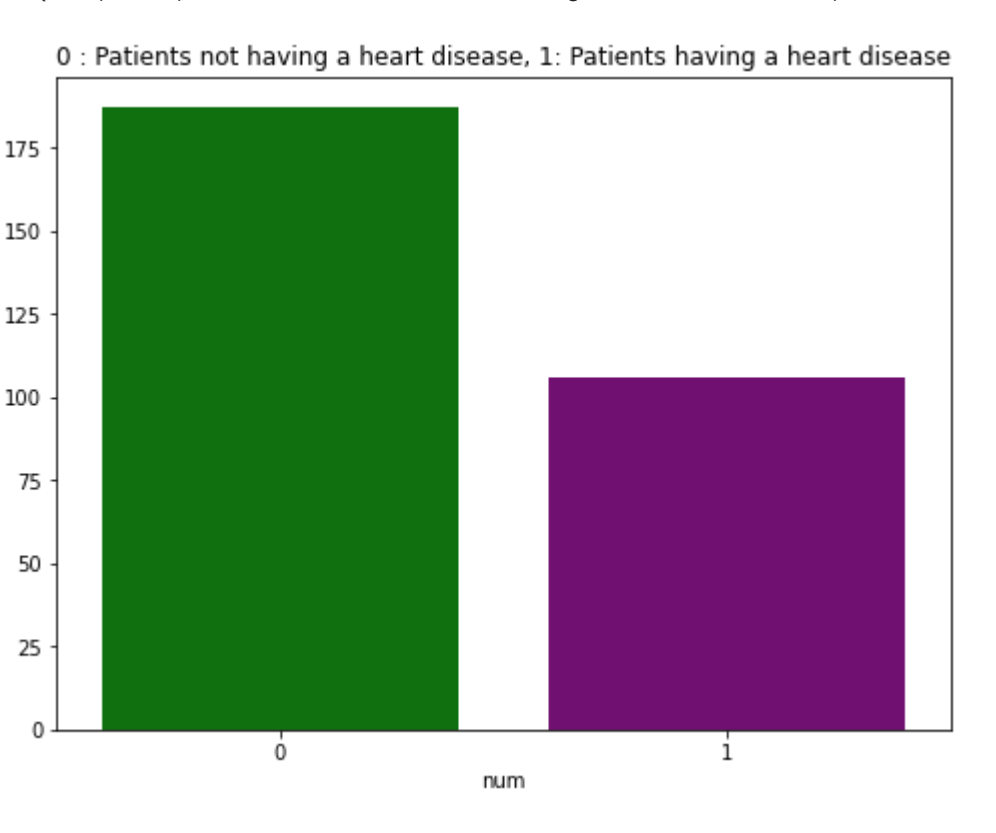
Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	num
count	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000	293.000000
mean	47.822526	0.726962	2.986348	132.660695	250.848708	0.068259	0.218430	139.058463	0.303754	0.588055	0.361775
std	7.824875	0.446282	0.965049	17.576783	65.059056	0.252622	0.460868	23.558003	0.460665	0.909554	0.481336
min	28.000000	0.000000	1.000000	92.000000	85.000000	0.000000	0.000000	82.000000	0.000000	0.000000	0.000000
25%	42.000000	0.000000	2.000000	120.000000	211.000000	0.000000	0.000000	122.000000	0.000000	0.000000	0.000000
50%	49.000000	1.000000	3.000000	130.000000	248.000000	0.000000	0.000000	140.000000	0.000000	0.000000	0.000000
75%	54.000000	1.000000	4.000000	140.000000	277.000000	0.000000	0.000000	155.000000	1.000000	1.000000	1.000000
max	66.000000	1.000000	4.000000	200.000000	603.000000	1.000000	2.000000	190.000000	1.000000	5.000000	1.000000

In [8]:
```python
corr=dataset.corr()
```

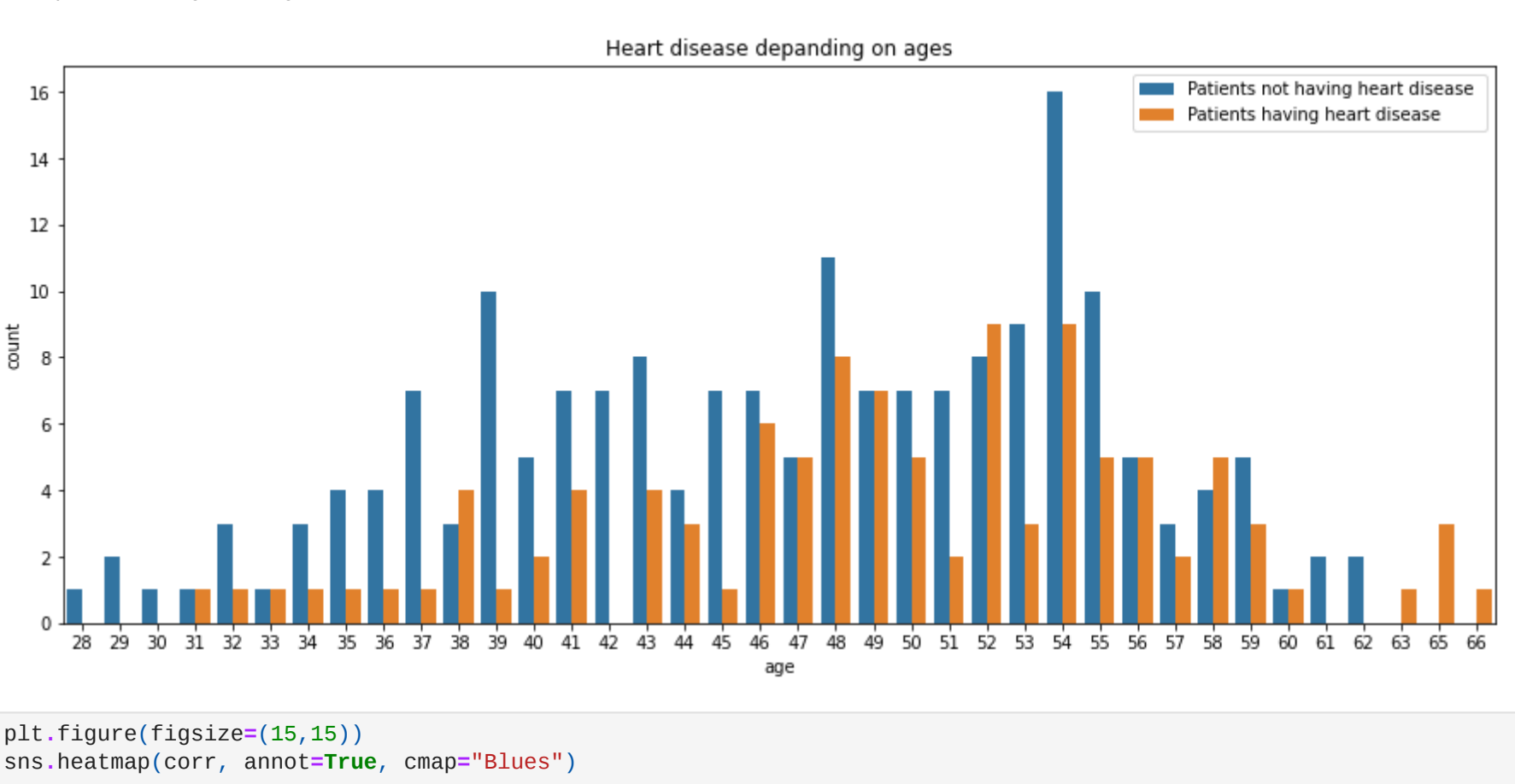
In [9]:
```python
plt.figure(figsize=(8,6))
sns.countplot(x=dataset.num, palette = ['green', 'purple'])
plt.title ("0 : Patients not having a heart disease, 1: Patients having a heart disease")
```

Out[9]: Text(0.5, 1.0, '0 : Patients not having a heart disease, 1: Patients having a heart disease')



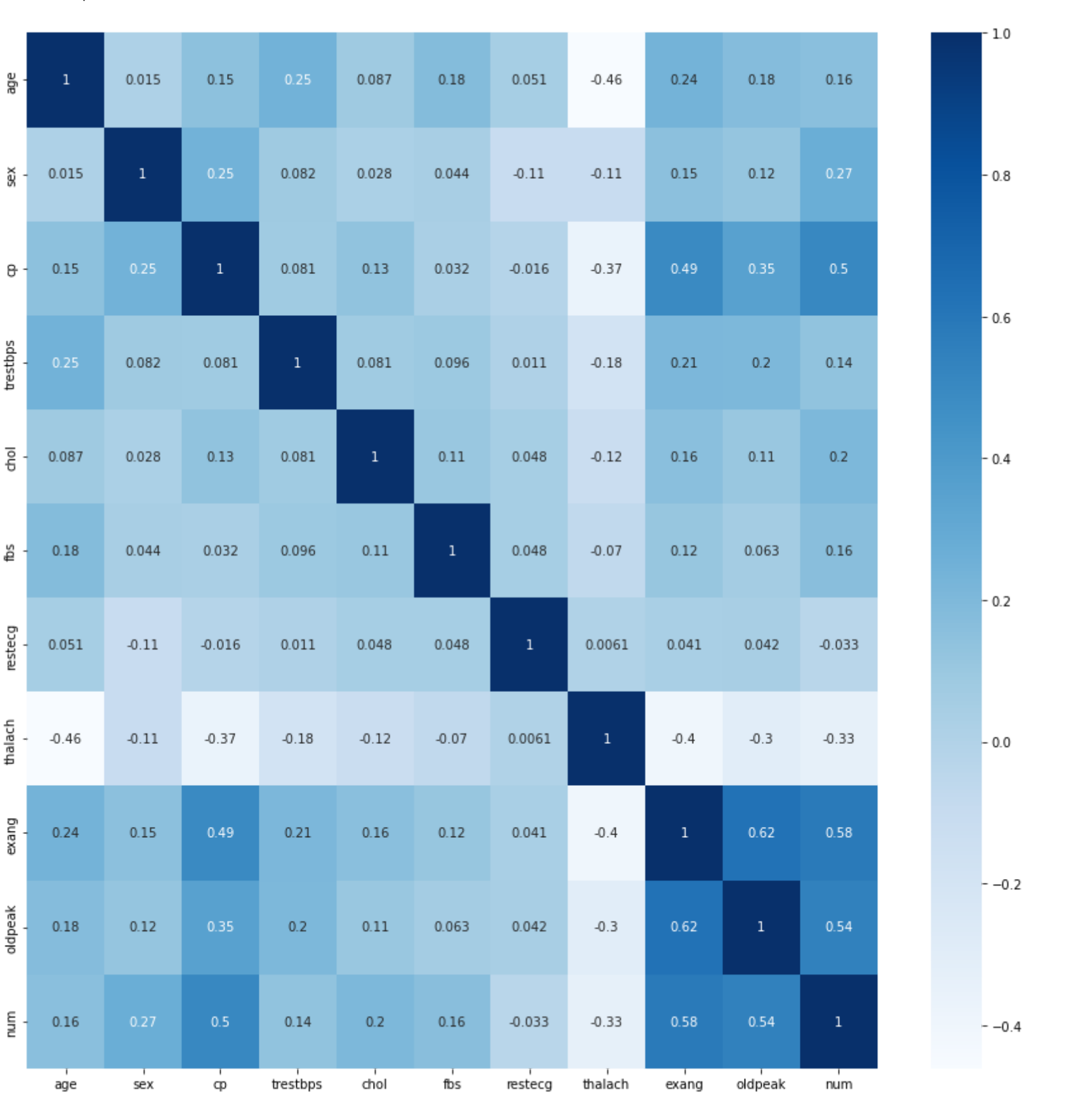
In [10]:
```python
plt.figure(figsize=(15,6))
sns.countplot(x = 'age', hue = 'num', data = dataset)
plt.title("Heart disease depanding on ages")
plt.legend(["Patients not having heart disease ","Patients having heart disease "], loc= "upper right")
```

Out[10]: <matplotlib.legend.Legend at 0x168a47944f0>



In [11]:
```python
plt.figure(figsize=(15,15))
sns.heatmap(corr, annot=True, cmap="Blues")
```

Out[11]: <AxesSubplot:>



LOGISTIC REGRESSION

In [12]:
```python
x = pd.DataFrame(dataset.iloc[:, :-3])
y = pd.DataFrame(dataset.iloc[:, -1])
```

In [13]:
```python
x.head()
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
0	28	1	2	130.0	132.000000	0.0	2.0	185.0
1	29	1	2	120.0	243.000000	0.0	0.0	160.0
2	29	1	2	140.0	250.848708	0.0	0.0	170.0
3	30	0	1	170.0	237.000000	0.0	1.0	170.0
4	31	0	2	100.0	219.000000	0.0	1.0	150.0

In [14]:
```python
y.head()
```

Out[14]:

	num
0	0
1	0
2	0
3	0
4	0

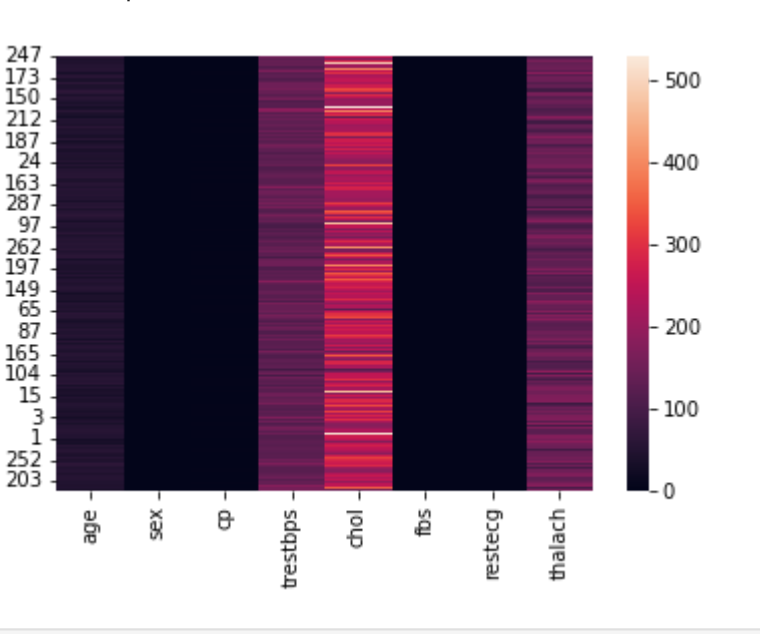
In [15]:
```python
x_train,x_test,y_train,y_test = train_test_split(x, y, test_size = 0.30, random_state = 1)
```

In [16]:
```python
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(205, 8)
(88, 8)
(205, 1)
(88, 1)
```

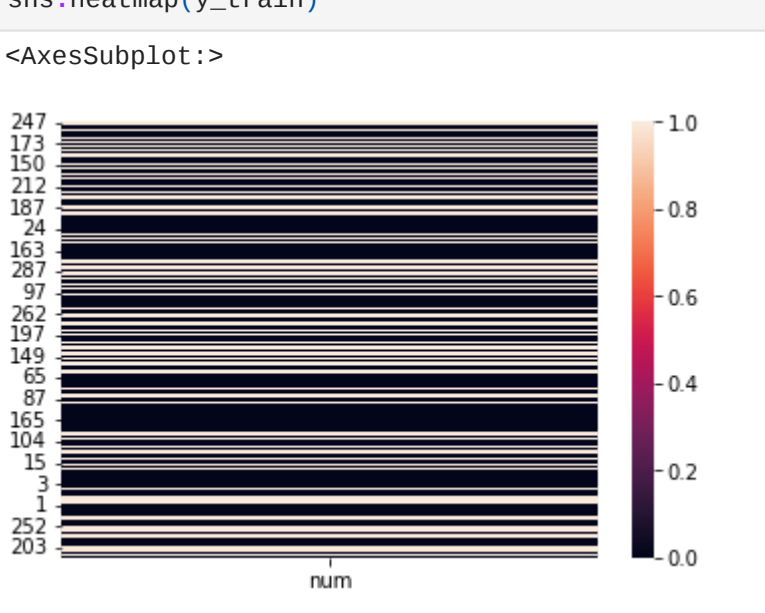
In [17]:
```python
sns.heatmap(x_train)
```

Out[17]: <AxesSubplot:>



In [18]:
```python
sns.heatmap(y_train)
```

Out[18]: <AxesSubplot:>



In [19]:
```python
model_1 = LogisticRegression(solver='lbfgs', max_iter=1000)
model_1.fit(x_train,y_train)
```

```
C:\Users\priyadharshan\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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)
```

Out[19]: LogisticRegression(max_iter=1000)

In [20]:
```python
y_pred_model_1 = model_1.predict(x_test)
```

In [21]:
```python
y_pred_model_1 = pd.DataFrame(y_pred_model_1, columns=['Predicted Values'])
y_pred_model_1.head()
```

Out[21]:

	Predicted Values
0	0
1	1
2	1
3	0
4	0

In [22]:
```python
conf_mat_model_1 = confusion_matrix (y_test, y_pred_model_1)
print (" Confusion Matrix for Logistic Regression Model: ")
conf_mat_model_1
```

```
 Confusion Matrix for Logistic Regression Model:
```

Out[22]:
```
array([[42, 16],
       [ 6, 24]], dtype=int64)
```

In [23]:
```python
accuracy_model_1 = accuracy_score (y_test, y_pred_model_1)
print ("Accuracy for Logistic Regression Model: ")
accuracy_model_1
```

```
Accuracy for Logistic Regression Model:
```

Out[23]: 0.75

In [24]:
```python
#RANDOM FOREST
```

In [25]:
```python
model_3 = RandomForestClassifier()
model_3.fit(x_train,y_train)
```

```
C:\Users\priyadharshan\AppData\Local\Temp\ipykernel_27048\13486048.py:2: 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().
  model_3.fit(x_train,y_train)
```

Out[25]: RandomForestClassifier()

In [26]:
```python
y_pred_model_3 = model_3.predict(x_test)
```

In [27]:
```python
y_pred_model_3 = pd.DataFrame(y_pred_model_3,columns=['Predicted Values'])
y_pred_model_3.head()
```

Out[27]:

	Predicted Values
0	0
1	1
2	1
3	0
4	0

In [28]:
```python
conf_mat_model_3 = confusion_matrix (y_test, y_pred_model_3)
print (" Confusion Matrix for Random Forest Model: ")
conf_mat_model_3
```

```
 Confusion Matrix for Random Forest Model:
```

Out[28]:
```
array([[44, 14],
       [ 7, 23]], dtype=int64)
```

In [29]:
```python
accuracy_model_3 = accuracy_score (y_test, y_pred_model_3)
print ("Accuracy for Random Forest Model: ")
accuracy_model_3
```

```
Accuracy for Random Forest Model:
```

Out[29]: 0.7613636363636364

In [30]:
```python
#DECISION TREE
```

In [31]:
```python
model_2 = DecisionTreeClassifier()
history=model_2.fit (x_train, y_train)
```

In [32]:
```python
y_pred_model_2 = model_2.predict (x_test)
```

In [33]:
```python
y_pred_model_2 = pd.DataFrame(y_pred_model_2,columns= ['Predicted Values'])
y_pred_model_2.head()
```

Out[33]:

	Predicted Values
0	0
1	1
2	1
3	0
4	0

In [34]:
```python
conf_mat_model_2 = confusion_matrix (y_test, y_pred_model_2)
print (" Confusion Matrix for Decission Tree Model: ")
conf_mat_model_2
```

```
 Confusion Matrix for Decission Tree Model:
```

Out[34]:
```
array([[41, 17],
       [ 7, 23]], dtype=int64)
```

In [35]:
```python
accuracy_model_2 = accuracy_score (y_test, y_pred_model_2)
print ("Accuracy for Decision Tree Model: ")
accuracy_model_2
```

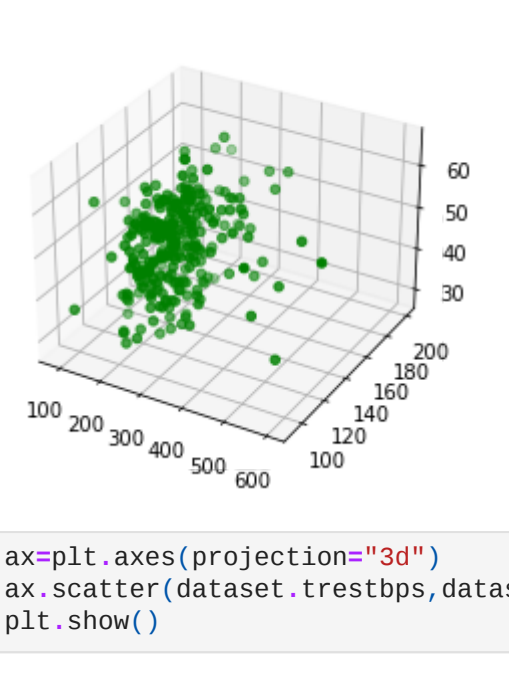
```
Accuracy for Decision Tree Model:
```

Out[35]: 0.7272727272727273

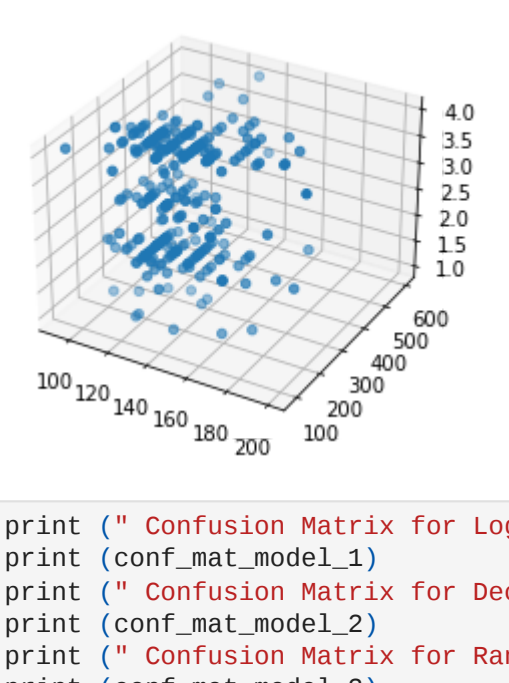
In [36]:
```python
#BEST MODEL
```

In [39]:
```python
ax=plt.axes(projection="3d")
ax.scatter(dataset.chol,dataset.trestbps,dataset.age,color='green')
plt.show
```

Out[39]: <function matplotlib.pyplot.show(close=None, block=None)>



In [47]:
```python
ax=plt.axes(projection="3d")
ax.scatter(dataset.trestbps,dataset.chol,dataset.cp)
plt.show()
```



In [48]:
```python
print (" Confusion Matrix for Logistic Regression: ")
print (conf_mat_model_1)
print (" Confusion Matrix for Decision Tree: ")
print (conf_mat_model_2)
print (" Confusion Matrix for Random Forest: ")
print (conf_mat_model_3)
```

```
 Confusion Matrix for Logistic Regression:
[[42 16]
 [ 6 24]]
 Confusion Matrix for Decision Tree:
[[41 17]
 [ 7 23]]
 Confusion Matrix for Random Forest:
[[44 14]
 [ 7 23]]
```

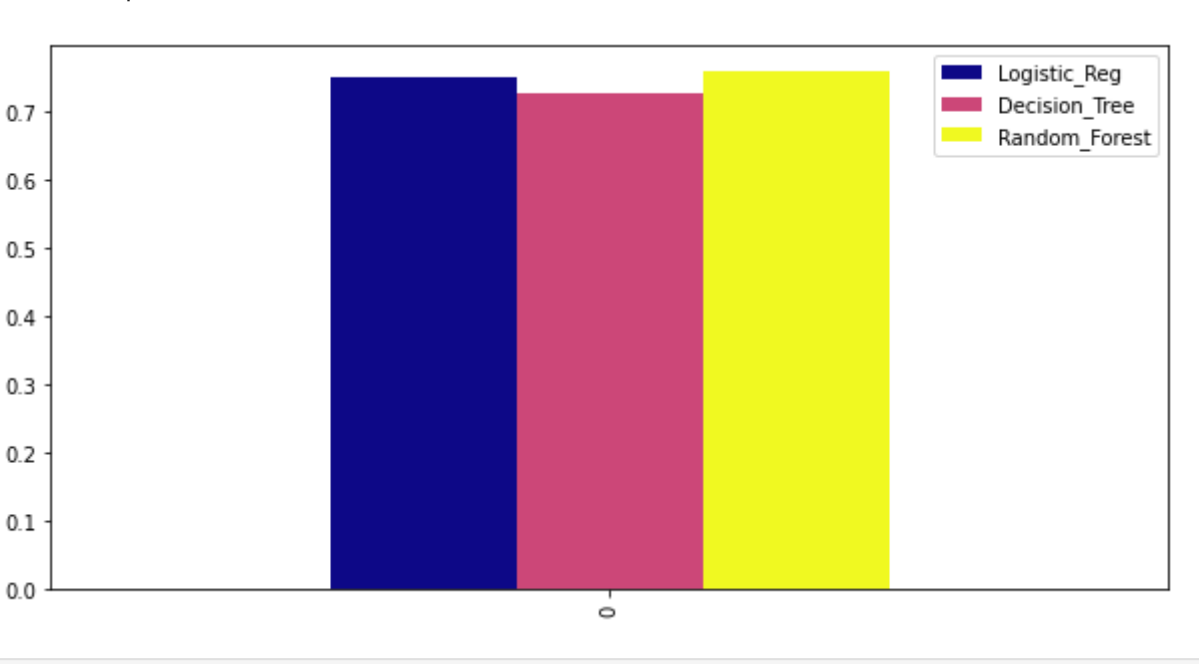
In [49]:
```python
score = { 'Logistic_Reg': [accuracy_model_1] , 'Decision_Tree': [accuracy_model_2], 'Random_Forest': [accuracy_model_3]}
score_df = pd.DataFrame(score)
score_df
```

Out[49]:

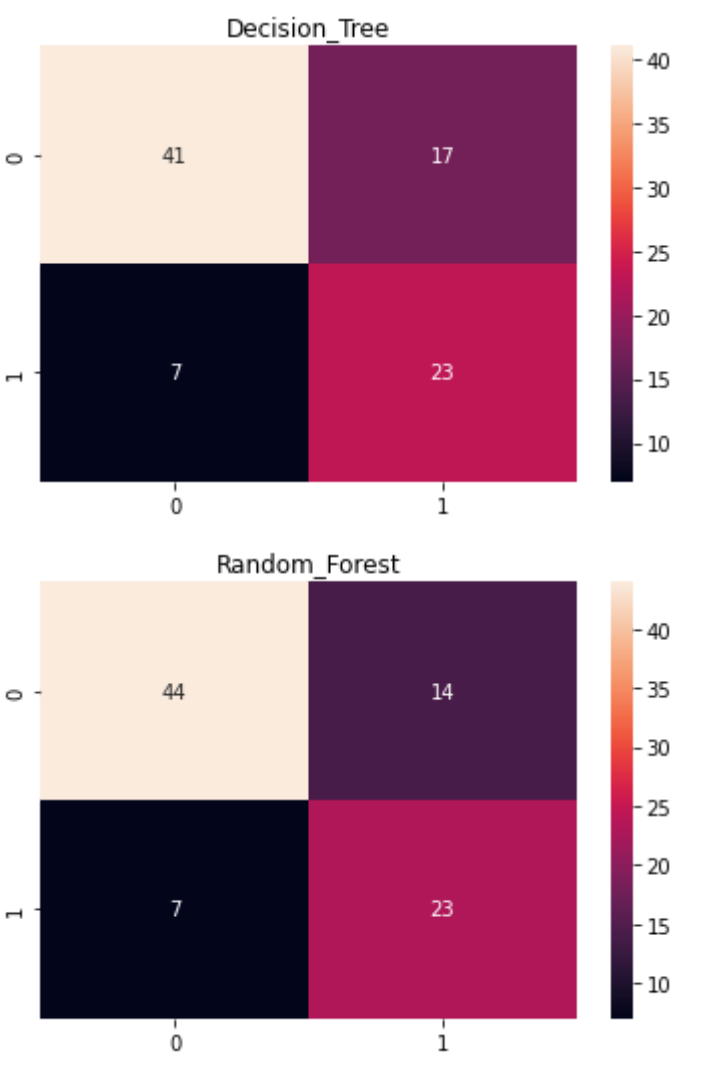
	Logistic_Reg	Decision_Tree	Random_Forest
0	0.75	0.727273	0.761364

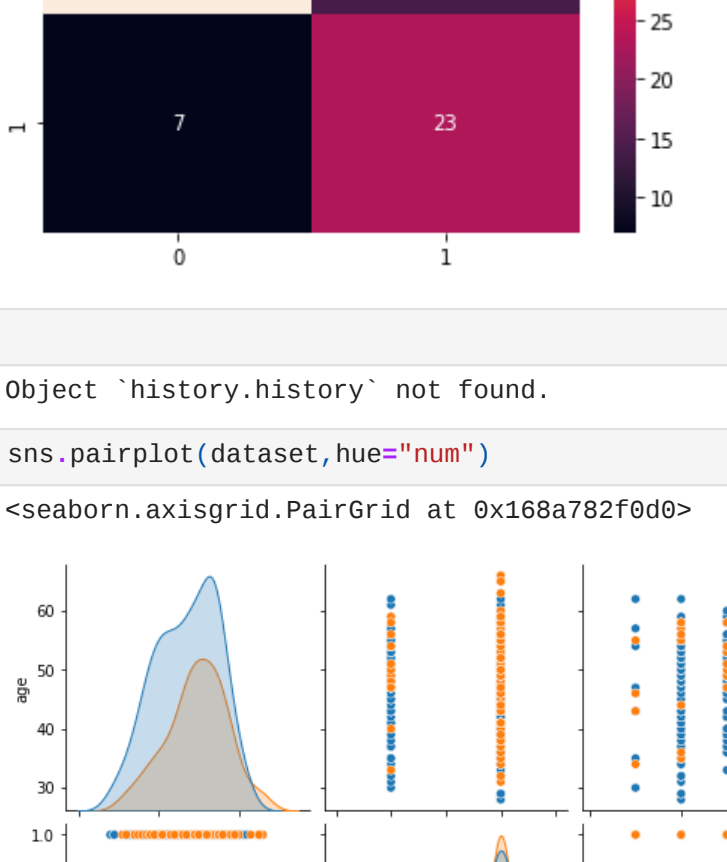
In [50]:
```python
score_df.plot(kind='bar', figsize=(10, 5), colormap='plasma')
```

Out[50]: <AxesSubplot:>



In [51]:
```python
conf_mat_final = {'Logistic_Reg': conf_mat_model_1, 'Decision_Tree': conf_mat_model_2, 'Random_Forest': conf_mat_model_3 }
for label,matrix in conf_mat_final.items():
    plt.title (label)
    sns.heatmap(matrix, annot=True)
    plt.show()
```



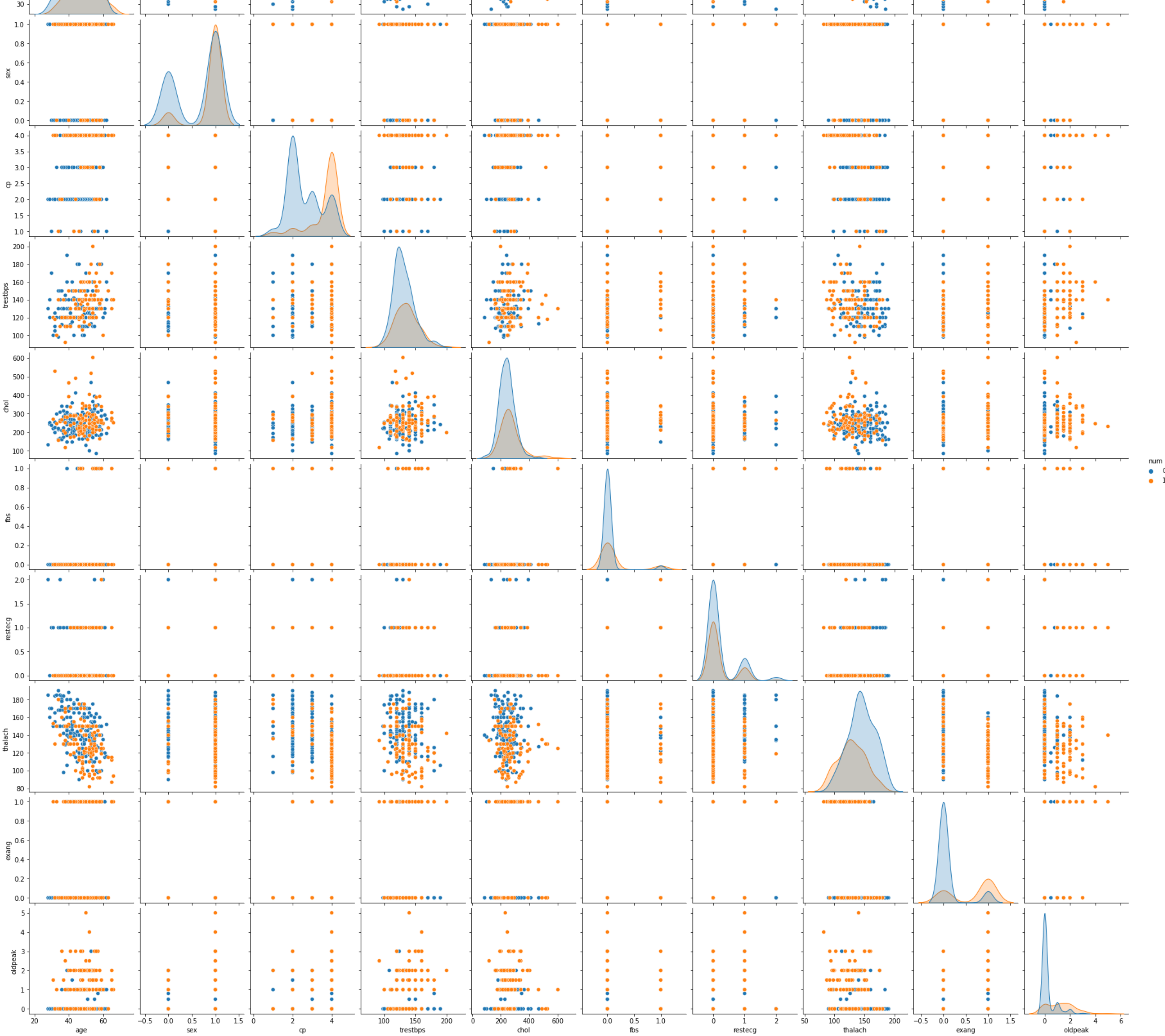


In [34]:

```
Object `history.history` not found.
```

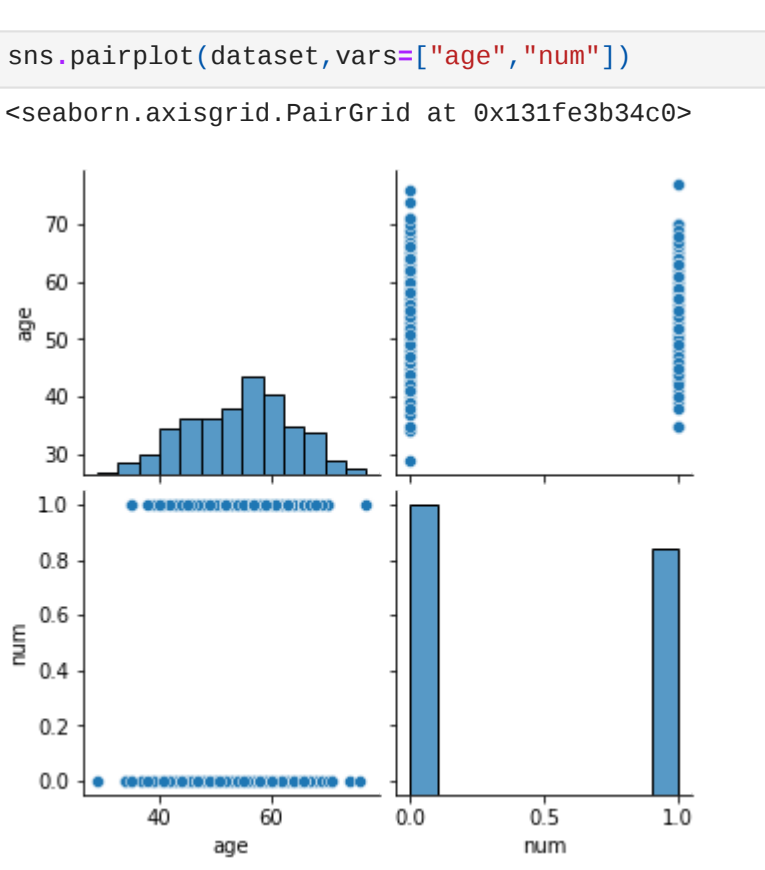
In [52]:
```python
sns.pairplot(dataset,hue="num")
```

Out[52]: <seaborn.axisgrid.PairGrid at 0x168a782f0d0>



In [62]:
```python
sns.pairplot(dataset,vars=["age","num"])
```

Out[62]: <seaborn.axisgrid.PairGrid at 0x131fe3b34c0>



In [ ]:

In [ ]: