```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pydotplus
from sklearn.tree import
DecisionTreeClassifier
import pandas as pd
from numpy import loadtxt
from keras.models import
Sequential
from keras.layers import Dense


# In[2]:


df=pd.read_csv('ML heart faliure
.csv')
df.info()
#Input variables: age, anemia, creatinine phosphokinase, diabetes, ejection
fraction,-
#-high blood pressure, platelets,serum creatinine, serum sodium, sex,
smoker
#Output Variables (y): Class Variable(0 or 1)


# In[3]:


dataset = loadtxt('ML heart
faliure .csv', delimiter=',')
X = dataset[:,0:12]
y = dataset[:,12]


# In[4]:


#Keras model,
Dense class for fully connected layers
model = Sequential()
model.add(Dense(12, input_dim=12,
activation='relu'))
model.add(Dense(12, activation='relu')) #using relu since tanh less
accuracy
model.add(Dense(1, activation='sigmoid'))


#
In[5]:


model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.fit(X, y, epochs=200, batch_size=5)


# In[6]:


_, accuracy =
model.evaluate(X, y)
print('Accuracy: %.2f' % (accuracy*100))


# In[7]:


#Probability
Predictions
predictions = model.predict(X)
```

```
#Rounding
rounded = [round(x[0]) for x in
predictions]
rounded


# In[8]:


predictions = (model.predict(X) >
0.5).astype(int)
predictions


# In[9]:


for i in range(299):
    print('%s => %d (expected
%d)' % (X[i].tolist(), predictions[i], y[i]))


# In[13]:


#descision tree
clf =
DecisionTreeClassifier(random_state=1234)
model = clf.fit(X, y)


# In[14]:


#textual
representation
text_representation = tree.export_text(clf)
print(text_representation)


# In[
]:


tree.plot_tree(clf)


# In[ ]:
```