import sklearn  
from sklearn.datasets import load\_breast\_cancer  
data = load\_breast\_cancer()  
label\_names = data['target\_names']  
labels = data['target']  
feature\_names = data['feature\_names']  
features = data['data']  
print(label\_names)  
print(labels[0])  
print(feature\_names[0])  
mean radius  
print(features[0])  
from sklearn.model\_selection import train\_test\_split  
train, test, train\_labels, test\_labels = train\_test\_split(features,labels,test\_size = 0.40, random\_state = 42)  
from sklearn.naive\_bayes import GaussianNB  
gnb = GaussianNB()  
model = gnb.fit(train, train\_labels)  
preds = gnb.predict(test)  
print(preds)  
from sklearn.metrics import accuracy\_score  
print(accuracy\_score(test\_labels,preds))

import sklearn  
from sklearn.datasets import load\_breast\_cancer  
data = load\_breast\_cancer()  
label\_names = data['target\_names']  
labels = data['target']  
feature\_names = data['feature\_names']  
features = data['data']  
print(label\_names)  
print(labels[0])  
print(feature\_names[0])  
mean radius  
print(features[0])  
from sklearn.model\_selection import train\_test\_split  
train, test, train\_labels, test\_labels = train\_test\_split(features,labels,test\_size = 0.40, random\_state = 42)  
from sklearn.naive\_bayes import GaussianNB  
gnb = GaussianNB()  
model = gnb.fit(train, train\_labels)  
preds = gnb.predict(test)  
print(preds)  
from sklearn.metrics import accuracy\_score  
print(accuracy\_score(test\_labels,preds))

import pydotplus  
from sklearn import tree  
from sklearn.datasets import load\_iris  
from sklearn.metrics import classification\_report  
from sklearn import cross\_validation  
import collections  
X = [[165,19] ,[175,32] ,[136,35] ,[174,65] ,[141,28] ,[176,15] ,[131,32] ,[166,6] ,[128,32] , [179,10], [136,34], [186,2] ,[126,25] ,[176,28] ,[112,38], [169,9], [171,36] ,[116,25] ,[196,25]]  
Y = ['Man', 'Woman' ,'Woman' ,'Man' ,'Woman' ,'Man' ,'Woman' ,'Man' ,'Woman', 'Man', 'Woman', 'Man' ,'Woman' ,'Woman' ,'Woman' ,'Man' ,'Woman' ,'Woman' ,'Man']data\_feature\_names = ['height','length of hair']  
X\_train, X\_test, Y\_train, Y\_test = cross\_validation.train\_test\_split  
(X, Y, test\_size=0.40, random\_state=5)  
clf = tree.DecisionTreeClassifier()  
clf = clf.fit(X,Y)  
prediction = clf.predict([[133,37]])  
print(prediction)  
dot\_data = tree.export\_graphviz(clf,feature\_names = data\_feature\_names, out\_file = None ,filled = True ,rounded = True)  
graph = pydotplus.graph\_from\_dot\_data(dot\_data)  
colors = ('orange', 'yellow')  
edges = collections.defaultdict(list)  
for edge in graph.get\_edge\_list():  
edges[edge.get\_source()].append(int(edge.get\_destination()))  
for edge in edges: edges[edge].sort()  
for i in range(2):dest = graph.get\_node(str(edges[edge][i]))[0]  
dest.set\_fillcolor(colors[i])  
graph.write\_png('Decisiontree16.png')

import pydotplus  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.model\_selection import train\_test\_split  
from sklearn.datasets import load\_breast\_cancer  
cancer = load\_breast\_cancer()  
import matplotlib.pyplot as plt  
import numpy as np  
cancer = load\_breast\_cancer()  
X\_train, X\_test, y\_train,  
y\_test = train\_test\_split(cancer.data, cancer.target, random\_state = 0)  
forest = RandomForestClassifier(n\_estimators = 50, random\_state = 0)  
forest.fit(X\_train,y\_train)  
print('Accuracy on the training subset:(:.3f)',format(forest.score(X\_train,y\_train)))  
print('Accuracy on the training subset:(:.3f)',format(forest.score(X\_test,y\_test)))  
n\_features = cancer.data.shape[1]  
plt.barh(range(n\_features),forest.feature\_importances\_, align='center')  
plt.yticks(np.arange(n\_features),cancer.feature\_names)  
plt.xlabel('Feature Importance')  
plt.ylabel('Feature')  
plt.show()