```
#creating the dataframe and the sample data
import pandas as pd
data = [['tom', 'boy', 10, 140], ['nick', 'boy', 15, 150], ['juli', 'girl', 14, 120], ['jim', 'boy', 16, 160], ['adam', 'boy', 17, 170], ['am
df = pd.DataFrame(data, columns=['Name', "sex", 'Age', 'weight'])
print(df)

#printing the shape, type, and column names
print(df.shape)
print(df.dtypes)
print(df.columns.tolist())
```

```
      Name   sex  Age  weight
0      tom   boy   10     140
1     nick   boy   15     150
2     juli  girl   14     120
3      jim   boy   16     160
4     adam   boy   17     170
5      amy  girl   23     110
6  wallace   boy    7      70
7  garrett   boy   28     200
(8, 4)
Name      object
sex       object
Age        int64
weight     int64
dtype: object
['Name', 'sex', 'Age', 'weight']
```

```
#creating array with dimension 6 by 6
import numpy as np

ini_array = np.array([[1, 2, 3, 6, 9, 0], [45, 4, 7, 12, 25, 64], [9, 6, 10, 11, 56, 96], [66, 86, 2, 50, 69, 98], [8, 55, 42, 9, 99, 22], [6

#printings shape and dimension of array
print(str(ini_array))
print(ini_array.shape)
print(ini_array.ndim)

#printing all the elements greater than the element at (3,3) which is 10
final_array = ini_array[(ini_array>10)]

print(final_array)
```

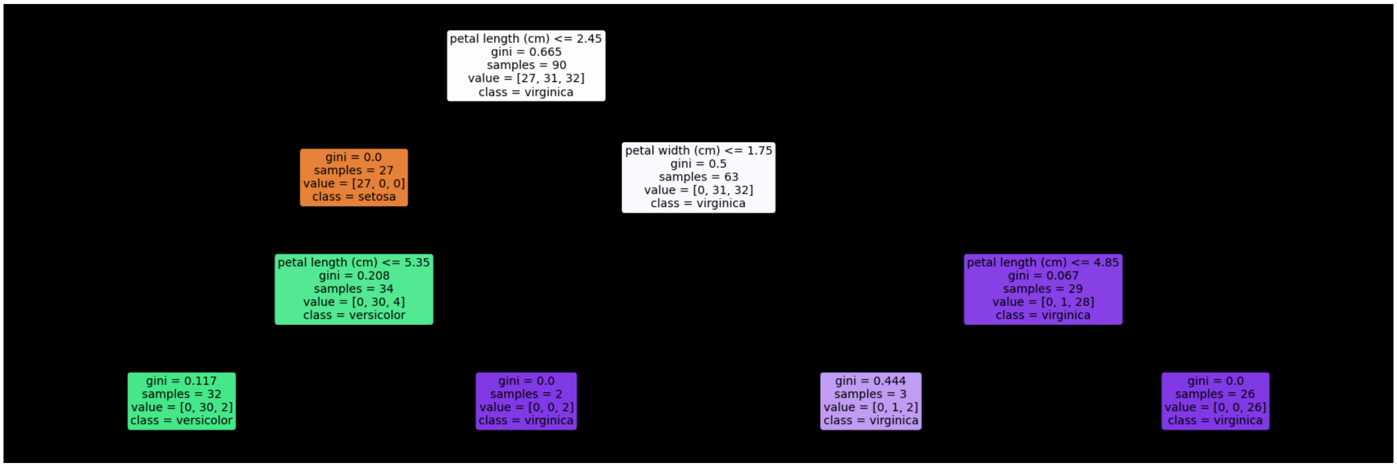
```
[[ 1  2  3  6  9  0]
 [45  4  7 12 25 64]
 [ 9  6 10 11 56 96]
 [66 86  2 50 69 98]
 [ 8 55 42  9 99 22]
 [68 89 33 29 67 68]]
(6, 6)
2
[45 12 25 64 11 56 96 66 86 50 69 98 55 42 99 22 68 89 33 29 67 68]
```

```
from sklearn.datasets import load_iris
from sklearn.metrics import confusion_matrix, classification_report

#load in the data
data = load_iris()
#convert to a dataframe
df = pd.DataFrame(data.data, columns = data.feature_names)
#create the species column
df['Species'] = data.target
#replace this with the actual names
target = np.unique(data.target)
target_names = np.unique(data.target_names)
targets = dict(zip(target, target_names))
df['Species'] = df['Species'].replace(targets)
#extract features and target variables
x = df.drop(columns="Species")
y = df["Species"]
#save the feature name and target variables
feature_names = x.columns
labels = y.unique()
#split the dataset
from sklearn.model_selection import train_test_split
X_train, test_x, y_train, test_lab = train_test_split(x,y,
                                                 test_size = 0.4,
                                                 random_state = 42)
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth =3, random_state = 42)
clf.fit(X_train, y_train)
#import relevant packages
from sklearn import tree
import matplotlib.pyplot as plt
#plt the figure, setting a black background
plt.figure(figsize=(30,10), facecolor ='k')
#create the tree plot
a = tree.plot_tree(clf,
                   #use the feature names stored
                   feature_names = feature_names,
                   #use the class names stored
                   class_names = labels,
                   rounded = True,
                   filled = True,
                   fontsize=14)
#show the plot
plt.show()

a(confusion_matrix(y_test, y_predict), annot = True)

print(classification_report(yi_test, yi_predict))
```



```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-27-bbe97349435b> in <module>
     44 plt.show()
     45
---> 46 a(confusion_matrix(y_test, y_predict), annot = True)

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

SEARCH STACK OVERFLOW

0s completed at 10:59 PM