

```
#Instalamos la libreria para python y paqueteria de graphviz
```

```
!pip install graphviz
```

```
!apt-get install graphviz
```

```
from sklearn.tree import DecisionTreeClassifier, export_graphviz
```

```
from sklearn.datasets import load_iris
```

```
from sklearn.model_selection import train_test_split
```

```
import graphviz
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
#Cargamos el dataset de iris
```

```
iris=load_iris()
```

```
#Separamos el dataset en entrenamiento y prueba
```

```
X_train, X_test, Y_train, Y_test = train_test_split(iris.data, iris.target)
```

```
#Definimos el modelo de arbol de decision, con todas los parametros por defecto
```

```
tree = DecisionTreeClassifier()
```

```
#Entrenamos el modelo
```

```
tree.fit(X_train, Y_train)
```

```
#Monstramos la presicion del modelo con el set de datos de prueba
```

```
tree.score(X_test, Y_test)
```

```
0.9736842105263158
```

```
#Mostramos la presicion del modelo con el set de datos para entrenar
```

```
tree.score(X_train, Y_train)
```

```
1.0
```

```
#Guardamos el arbol en un archivo .dot para mostrarlo a continuacion
```

```
export_graphviz(tree, out_file='tree.dot', class_names=iris.target_names,  
                feature_names=iris.feature_names, impurity=False, filled=True)
```

```
#Mostramos el arbol de desicion
```

```
with open('tree.dot') as f:
```

```
    dot_graph=f.read()
```

```
graphviz.Source(dot_graph)
```

```
#Mostramos la importancia de cada caracteristica del dataset
```

```
carac = iris.data.shape[1]
```

```
plt.barh(range(carac), tree.feature_importances_)
```

```
plt.yticks(np.arange(carac), iris.feature_names)
```

```
plt.xlabel('Importancia de las caracteristicas')
```

```
plt.ylabel('Caracteristica')
```

```
plt.show()
```