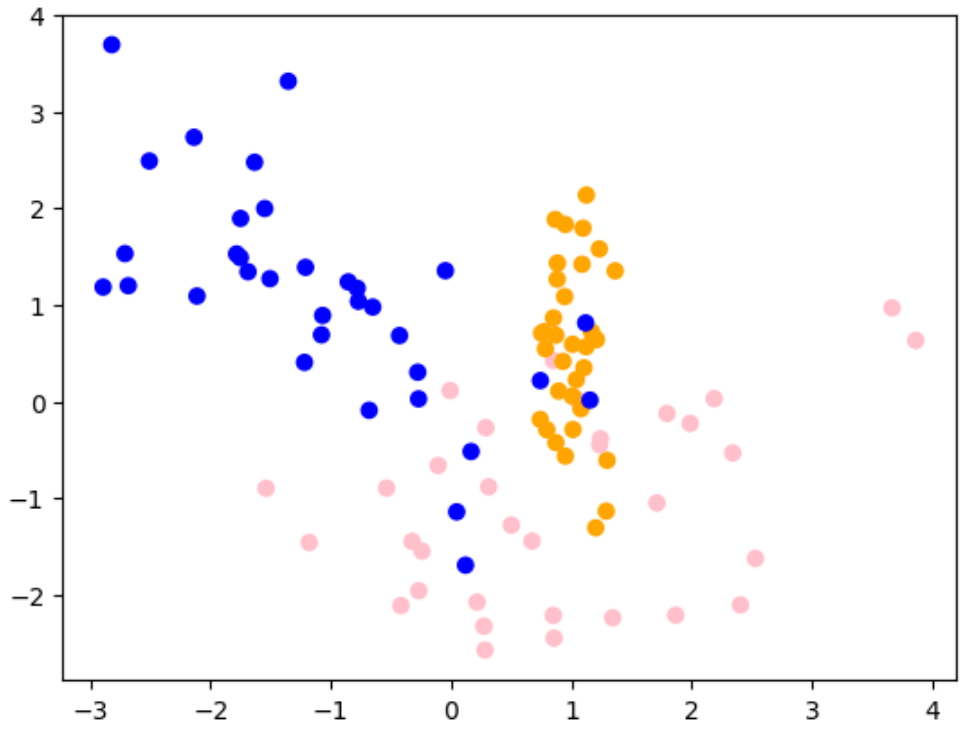
In [3]:
```
import sklearn
import numpy as np
from sklearn import datasets
from sklearn.datasets import make_classification
```

In [4]:
```
X, y = make_classification(n_samples=100, n_features=2, n_classes=3,
                           n_redundant=0, n_informative=2,
                           n_clusters_per_class=1, random_state=0)
```

In [7]:
```
import matplotlib.pyplot as plt
colors = np.array(["blue", "orange", "pink"])
plt.scatter(X[:,0], X[:, 1], color=colors[y])
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x7f351b0d1f10>
```



In [9]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [12]:
```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(max_leaf_nodes=10)
dt.fit(X_train, y_train)
```

Out[12]:
```
DecisionTreeClassifier
DecisionTreeClassifier(max_leaf_nodes=10)
```

In [13]:
```
dt.score(X_test, y_test)
```

Out[13]:
```
0.65
```

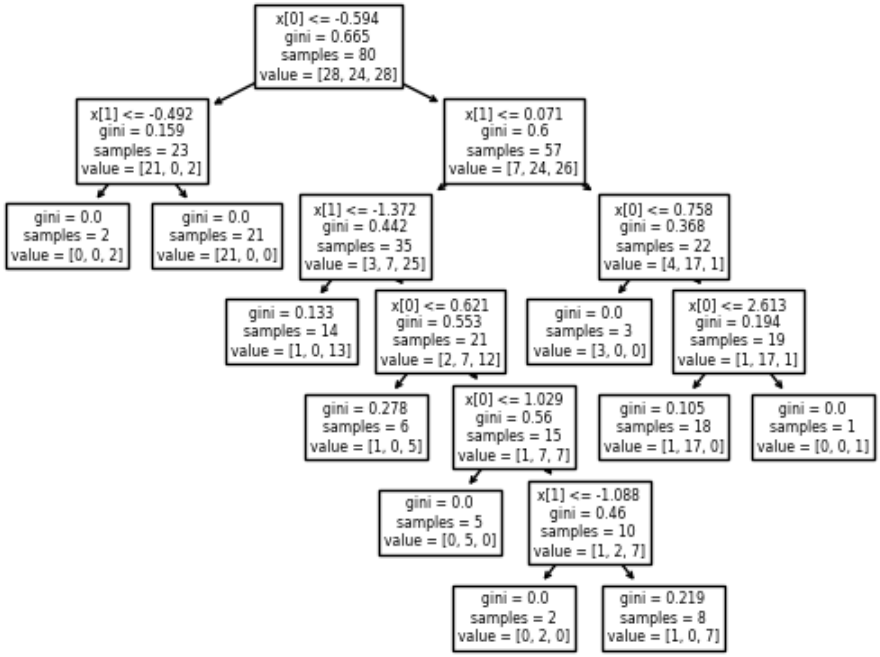
In [14]:
```
dt.get_depth()
```

Out[14]:
```
6
```

In [16]:
```
dt.predict_proba(X_test[-2:])
```

Out[16]:
```
array([[0.16666667, 0.        , 0.83333333],
       [1.        , 0.        , 0.        ]])
```

In [18]:
```
from sklearn import tree
tree.plot_tree(dt)
plt.show()
```



In [19]:
```
tr = dt.tree_ j# tree object
```

In [23]:
```
# root node 0
tr.feature[0], tr.threshold[0], tr.impurity[0], tr.value[0]
```

Out[23]:
```
(0, -0.5935284197330475, 0.665, array([[28., 24., 28.]]))
```

In [25]:
```
# left child of root node 0
left_child_index = tr.children_left[0]
tr.threshold[left_child_index]
```

Out[25]:
```
-0.4916086085140705
```

In [26]:
```
dt.feature_importances_
```

Out[26]:
```
array([0.58489068, 0.41510932])
```

In [27]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=3, random_state=0, max_features=1)
rf.fit(X_train, y_train)
```

Out[27]:
```
RandomForestClassifier
RandomForestClassifier(max_features=1, n_estimators=3, random_state=0)
```

In [28]:
```
param_grid = [
    {"n_estimators": np.arange(1,10),
     "max_features": [1,2]}
]
```