Decision Tree Classifier

In [1]:

```
from sklearn.datasets import load_iris
from sklearn import tree
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import train_test_split
```

In [2]:

```
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size = 0.4
```

In [3]:

```
clf_entropy = tree.DecisionTreeClassifier(random_state = 17, criterion = 'entropy', min_sam
clf = clf_entropy.fit(X_train, y_train)

y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred, target_names = iris.target_names))
print('\nAccuracy: {0:.4f}'.format(accuracy_score(y_test, y_pred)))
```

```
             precision    recall  f1-score   support

     setosa       1.00      1.00      1.00        15
 versicolor       0.96      0.93      0.94        27
  virginica       0.89      0.94      0.92        18

avg / total       0.95      0.95      0.95        60


Accuracy: 0.9500
```

In [4]:

```
from sklearn.metrics import confusion_matrix
C = confusion_matrix(y_test, clf_entropy.predict(X_test))
print(C)
print()
print(C.sum(axis=1))
```

```
[[15  0  0]
 [ 0 25  2]
 [ 0  1 17]]

[15 27 18]
```

In [5]:

```
from sklearn.tree import export_graphviz
from io import StringIO
from IPython.display import Image
import pydotplus
import graphviz
import pydot
```

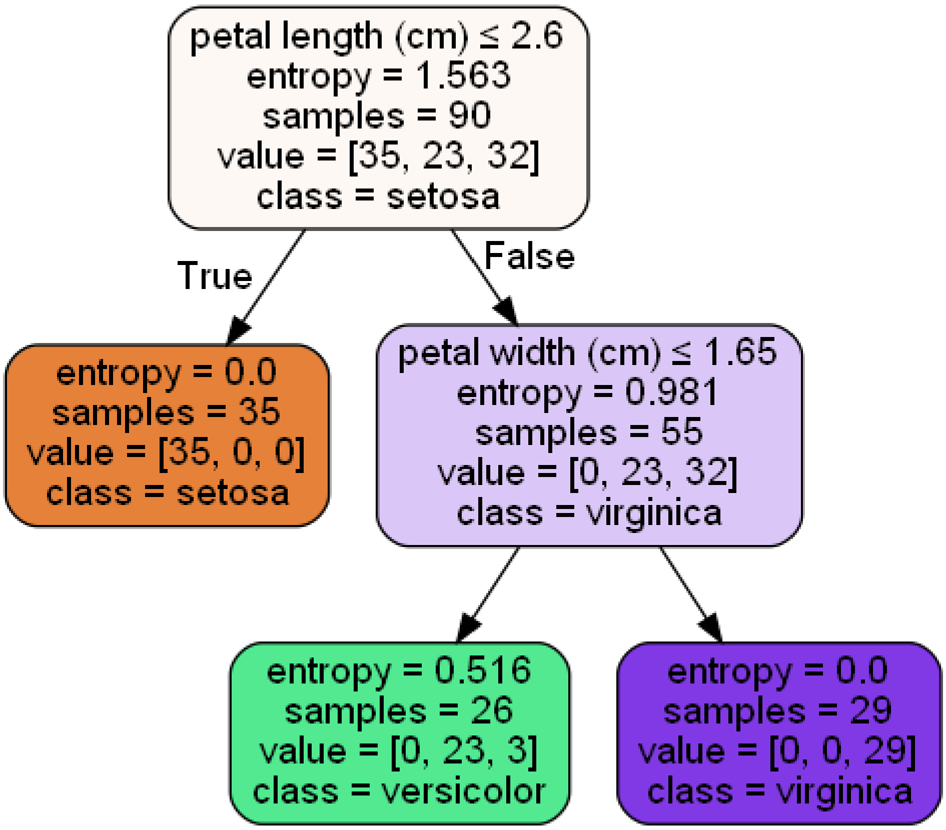
In [6]:

```
dot_data = StringIO()
tree.export_graphviz(clf, out_file=dot_data, feature_names=iris.feature_names, class_names=


graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())

#graph.write_png('D_tree.png')
```

Out[6]:



In [ ]: