```
from sklearn import tree
import csv
import numpy as np
import matplotlib.pyplot as plt

f = 'data/kaggle_train_tf_idf.csv'

with open(f, 'r') as fin:
    data = np.array(list(csv.reader(fin)))

NUM_TRAININGS = 3000
X_train = data[1:NUM_TRAININGS, 1:-1]
Y_train = data[1:NUM_TRAININGS, -1]
X_test = data[NUM_TRAININGS:, 1:-1]
Y_test = data[NUM_TRAININGS:, -1]
X = data[1:, 1:-1]
Y = data[1:, -1]


def get_error(G,Y):
    error = 0
    for i in xrange(len(G)):
        if G[i] != Y[i]:
            error += 1

    return 1.0 * error / len(G)


min_samples_leafs = [i for i in range(1,26)]
test_errors = []
train_errors = []

min_test_error = float("inf")

for min_samples_leaf in min_samples_leafs:
    # initialize the tree model
    clf = tree.DecisionTreeClassifier(criterion='entropy', min_samples_leaf=min_samples_leaf)
    # train the model
    clf = clf.fit(X_train, Y_train)
    # make prediction
    G_train = clf.predict(X_train)
    G_test = clf.predict(X_test)

    # compute error
    train_error = get_error(G_train, Y_train)
    train_errors.append(train_error)
    test_error = get_error(G_test, Y_test)
    test_errors.append(test_error)


best_sample = min_samples_leafs[test_errors.index(min(test_errors))]
"""
plt.figure(1)
plt.plot(min_samples_leafs, train_errors)
plt.plot(min_samples_leafs, test_errors)
plt.xlabel('min_samples_leaf')
plt.ylabel('Error')
plt.title('Plot of Error vs. min_samples_leaf')
plt.legend(['train_error', 'test_error'])
plt.show()
"""

# increment by depth
```

```
max_depths = [i for i in range(2,21)]
test_errors = []
train_errors = []


for max_depth in max_depths:
    # initialize the tree model
    clf = tree.DecisionTreeClassifier(criterion='entropy', max_depth=max_depth)
    # train the model
    clf = clf.fit(X_train, Y_train)
    # make prediction
    G_train = clf.predict(X_train)
    G_test = clf.predict(X_test)

    # compute error
    train_error = get_error(G_train, Y_train)
    train_errors.append(train_error)
    test_error = get_error(G_test, Y_test)
    test_errors.append(test_error)


best_depth = max_depths[test_errors.index(min(test_errors))]
"""
plt.figure(2)
plt.plot(max_depths, train_errors)
plt.plot(max_depths, test_errors)
plt.xlabel('max_depths')
plt.ylabel('Error')
plt.title('Plot of Error vs. max_depth')
plt.legend(['train_error', 'test_error'])
plt.show()
"""

# train best model
clf_best = tree.DecisionTreeClassifier(criterion='entropy', max_depth=best_depth, min_samples_leaf=
best_sample)
# train the model
clf_best = clf_best.fit(X, Y)

# cross validation
print best_depth
print best_sample
K = 5
from sklearn import cross_validation
scores = cross_validation.cross_val_score(clf_best, X, Y, cv=K, scoring='accuracy')
avg_score = sum(scores) / len(scores)
print('Scores = {}'.format(scores))
print('avg_score = {}'.format(avg_score))
```