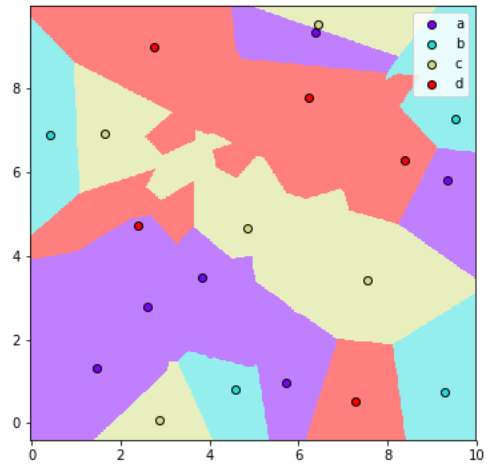
To run the test code in this document, you will need to place the file ClassificationPlotter.py into the same folder as the file KNNClassifier.py.

Example 1:

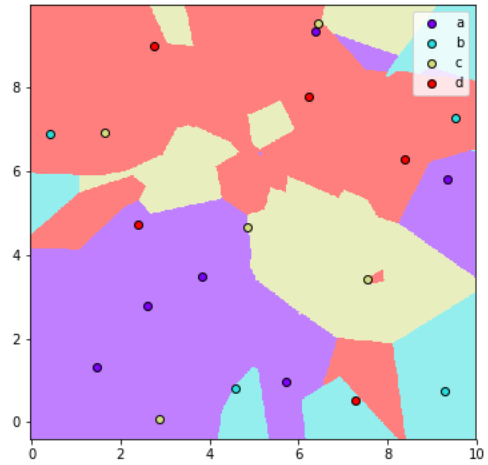
The code shown on the left below might take a few moments to run, but should generate the output shown on the right.

```
from ClassificationPlotter import plot_regions

np.random.seed(1204)
X = np.random.uniform(0,10,40).reshape(20,2)
y = np.random.choice(['a','b','c','d'],20)

knn_mod_3 = KNNClassifier(X,y,3)
print(knn_mod_3.score(X,y))
print(knn_mod_3.predict(X))

plot_regions(knn_mod_3, X, y, 500)

knn_mod_4 = KNNClassifier(X,y,4)
print(knn_mod_4.score(X,y))
print(knn_mod_4.predict(X))

plot_regions(knn_mod_4, X, y, 500)
```

```
0.95
['c' 'd' 'c' 'a' 'a' 'c' 'b' 'a' 'b' 'c'
'd' 'a' 'd' 'a' 'a' 'a' 'b' 'd' 'b' 'c']
```




```
0.55
['d' 'b' 'a' 'a' 'a' 'a' 'd' 'd' 'b' 'd'
'd' 'a' 'd' 'a' 'a' 'a' 'b' 'd' 'd' 'c']
```

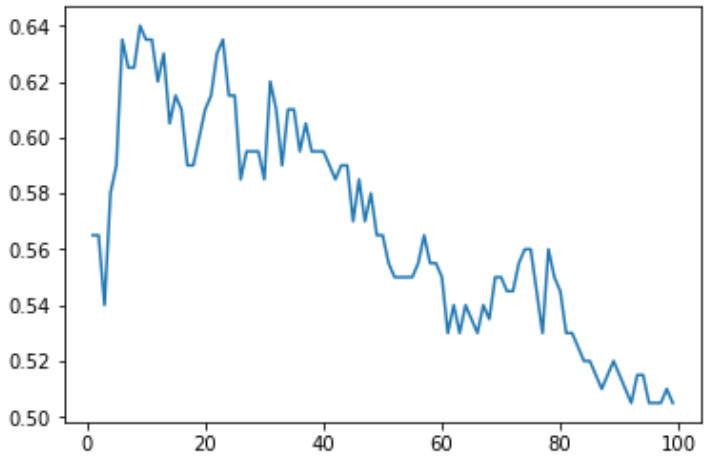


Example 2:

Consider the following code.

```
np.random.seed(1548)

from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

X, y = make_classification(n_samples=2000, n_features=10, n_informative=4,
                           n_clusters_per_class=1,class_sep=0.5,n_classes=6 )

X_train, X_holdout, y_train, y_holdout = train_test_split(X, y, test_size=0.2)
X_val, X_test, y_val, y_test = train_test_split(X_holdout, y_holdout,
                                                test_size=0.5)

rng = range(1,100)
val_acc = []

for K in rng:
    temp_mod = KNNClassifier(X_train, y_train, K)
    val_acc.append(temp_mod.score(X_val,y_val))

plt.close()
plt.plot(rng, val_acc)
plt.show()

knn_mod = KNNClassifier(X_train, y_train, 10)
print("Training Accuracy:", knn_mod.score(X_train, y_train))
print("Testing Accuracy: ", knn_mod.score(X_test, y_test))
print(knn_mod.predict(X_test[:20,:]))
print(y_test[:20])
```

The code above should generate the following output:




```
Training Accuracy: 0.72625
Testing Accuracy:  0.595
[1 1 2 4 1 4 2 4 2 0 4 5 0 1 2 5 0 2 0 3]
[3 1 3 1 1 4 2 4 2 5 5 3 0 1 0 5 1 2 1 2]
```