

1 数据集介绍：

虹膜



150个实例

萼片长度，萼片宽度，花瓣长度，花瓣宽度
(sepal length, sepal width, petal length and petal width)

类别：

Iris setosa, Iris versicolor, Iris virginica.



2. 利用Python的机器学习库sklearn: SkLearnExample.py

```
from sklearn import neighbors
from sklearn import datasets

knn = neighbors.KNeighborsClassifier()

iris = datasets.load_iris()

print iris

knn.fit(iris.data, iris.target)

predictedLabel = knn.predict([[0.1, 0.2, 0.3, 0.4]])

print predictedLabel
```

3. KNN 实现Implementation:

Example of kNN implemented from Scratch in Python

```
import csv
import random
import math
import operator
```

```
def loadDataset(filename, split, trainingSet=[], testSet=[]):
    with open(filename, 'rb') as csvfile:
        lines = csv.reader(csvfile)
        dataset = list(lines)
        for x in range(len(dataset)-1):
            for y in range(4):
                dataset[x][y] = float(dataset[x][y])
            if random.random() < split:
                trainingSet.append(dataset[x])
            else:
                testSet.append(dataset[x])
```

```
def euclideanDistance(instance1, instance2, length):
    distance = 0
    for x in range(length):
        distance += pow((instance1[x] - instance2[x]), 2)
    return math.sqrt(distance)
```

```
def getNeighbors(trainingSet, testInstance, k):
    distances = []
    length = len(testInstance)-1
    for x in range(len(trainingSet)):
        dist = euclideanDistance(testInstance, trainingSet[x], length)
        distances.append((trainingSet[x], dist))
    distances.sort(key=operator.itemgetter(1))
    neighbors = []
    for x in range(k):
        neighbors.append(distances[x][0])
    return neighbors
```

```
def getResponse(neighbors):
    classVotes = {}
    for x in range(len(neighbors)):
        response = neighbors[x][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
            classVotes[response] = 1
    sortedVotes = sorted(classVotes.iteritems(), key=operator.itemgetter(1), reverse=True)
    return sortedVotes[0][0]
```

```
def getAccuracy(testSet, predictions):
    correct = 0
    for x in range(len(testSet)):
        if testSet[x][-1] == predictions[x]:
            correct += 1
    return (correct/float(len(testSet))) * 100.0
```

```
def main():
    # prepare data
    trainingSet=[]
    testSet=[]
    split = 0.67
    loadDataset(r'D:\MaiziEdu\DeepLearningBasics_MachineLearning\Datasets\iris.data.txt', split, trainingSet, testSet)
```

```
print 'Train set: ' + repr(len(trainingSet))
print 'Test set: ' + repr(len(testSet))
# generate predictions
predictions=[]
k = 3
for x in range(len(testSet)):
    neighbors = getNeighbors(trainingSet, testSet[x], k)
    result = getResponse(neighbors)
    predictions.append(result)
    print('> predicted=' + repr(result) + ', actual=' + repr(testSet[x][-1]))
accuracy = getAccuracy(testSet, predictions)
print('Accuracy: ' + repr(accuracy) + '%')

main()
```