```
# -*- coding: utf-8 -*-
"""
Created on Wed Aug 19 14:28:12 2018

@author: Walter Stevens

Simple example of PCA (Principal Component Analyis)using the public Iris dataset.

"""

from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
import pylab as pl
from itertools import cycle

iris = load_iris()

numSamples, numFeatures = iris.data.shape
#print(numSamples)
#print(numFeatures)
#print(list(iris.target_names))

#distilling the4 dimensinos down into 2

X = iris.data
pca = PCA(n_components=2, whiten=True).fit(X)
X_pca = pca.transform(X)

#print(pca.components_)

#print(pca.explained_variance_ratio_)
```

```
#print(sum(pca.explained_variance_ratio_))

#[0.92461872 0.05306648]
#0.977685206318795

#Although we have discarded two of our four dimensions, PCA has chosen the
#remaining two dimensions so well that we've captured 92% of the variance in
#our data in a single dimension alone. The second dimension just gives an
#additional 5%; altogether by projecting it down to two dimensions, we've only
#really lost less than 3% of the variance in the data

#Plotting the resultant points.

%matplotlib inline
from pylab import *

colors = cycle('rgb')
target_ids = range(len(iris.target_names))
pl.figure()
for i, c, label in zip(target_ids, colors, iris.target_names):
    pl.scatter(X_pca[iris.target == i, 0], X_pca[iris.target == i, 1],
        c=c, label=label)
pl.legend()
pl.show()
```

#The three different types of Iris are still clustered fairly well.