Practical 1: Principal Component Analysis

In this practical, we will implement a simple unsupervised learning algorithm using the Breast Cancer Wisconsin Diagnostic Data Set. This dataset contains 569 samples with 30 numerical parameters each. Each sample is labeled either *malignant* or *benign*. We will use the first 400 samples to train the algorithm and we will test the results with the rest.

You have to implement the Principal Component Analysis (PCA) and apply it to the training dataset, so you reduce the dimension of each sample to 3. Compute the eigenvectors associated to the three highest eigenvalues using the algorithm sketched in Slide 11 of these lecture slides. Once you have reduced the dimensionality of your training data set, use the k-means algorithm (which we'll study later in the week) to cluster it. You can use a library function for the clustering step. The function can be loaded with the following:

```
from sklearn.cluster import KMeans
```

Set the number of clusters to 2, as we want to classify each sample into malignant or benign.

Apply the trained algorithm to the test data, that is, reduce its dimensionality with the trained PCA and apply clustering. Check the percentage of samples that have been clustered correctly. Note that the algorithm creates two clusters but the first cluster does not have to correspond to the first label.

You can automatically download and load the data with the following lines of code:

```
from sklearn.datasets import load_breast_cancer
dataset = load_breast_cancer()
```

The dataset contains an array with the samples in `dataset.data` and an array with the labels in `dataset.target`. The internal representation of the labels uses 0 for benign and 1 for malign.

Use `matplotlib` to plot the training data, after the dimensionality reduction. Use 4 different colours, points with different labels or in different clusters should be plotted with different colours.