1. **Support Vector Machines**

Load the Iris dataset from sklearn.

- Create a scatter plot of the dataset.

The points in the dataset are of 3 classes : Setosa, Versicolour, and Virginica. Split the dataset into train, val and test.

- Using your favorite library, train an SVM classifier to predict the class of a given datapoint. Report the accuracy, precision, recall and f1-score.
- Modify the parameters of the SVM to get a soft-margin as well as a hard-margin classifier. Compare the performances of the two. Tune the respective hyperparameters to get the best results on the validation set. Which parameter effects the hardness of the classifier?
- Apply the Gaussian, RBF and polynomial kernels to better classifier(hard-margin or soft-margin). Which kernel gives the best performance? You may need to tune the parameters of the kernels to get the best result for each.

2. **Singular Value Decomposition**

Load the following images from skimage

```
from skimage import data
from skimage.color import rgb2gray
from ipywidgets import interact,interactive,interact_manual
```

```
[ ] from skimage import img_as_ubyte,img_as_float
    gray_images = {
            "cat":rgb2gray(img_as_float(data.chelsea())),
            "astro":rgb2gray(img_as_float(data.astronaut())),
            "camera":data.camera(),
            "coin": data.coins(),
            "clock":data.clock(),
            "blobs":data.binary_blobs(),
            "coffee":rgb2gray(img_as_float(data.coffee()))
    }
```

Since an image is basically a matrix, we can compute the SVD of the matrix.

- Plot any one of these images using matplotlib and extract the data matrix for this image.
- Compute the SVD of the matrix, and compute a rank-k approximation of the image, where k is user-defined. This will be the reconstructed image

- For different values of k, plot the original image as well as the reconstructed image. How does the quality vary with k?
- Note that, if the original image is of size M x N, and we are computing a k rank approximation, then we need to store the M x k left singular matrix, the N x k right singular matrix as well as k singular values. What is the compression ratio with respect to the original image? Report the compression ratio for different values of k.

Repeat these 4 questions in case of color images. (How do you convert an RGB image, having shape M x N x 3 into a 2D matrix, so that you can apply SVD?)