Write a Python program to illustrate in 2D and in 3D the features' reduction of Iris dataset with the
following three dimensionality reduction methods:
1. Principal Component Analysis,
2. Isometric Mapping,
3. Locally linear embedding,
Which of the three features' reduction methods is relevant for the Iris dataset? Justify your answer.

PCA:

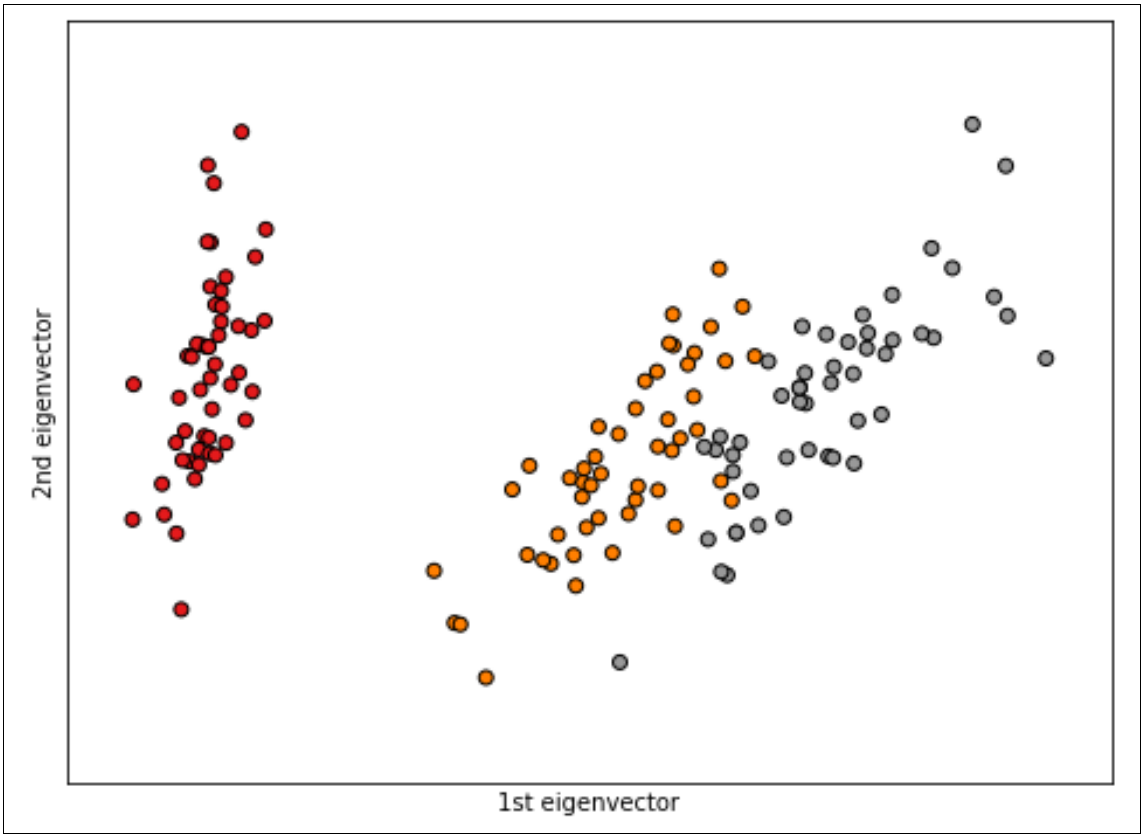

figure 3.1 -- 2D PCA

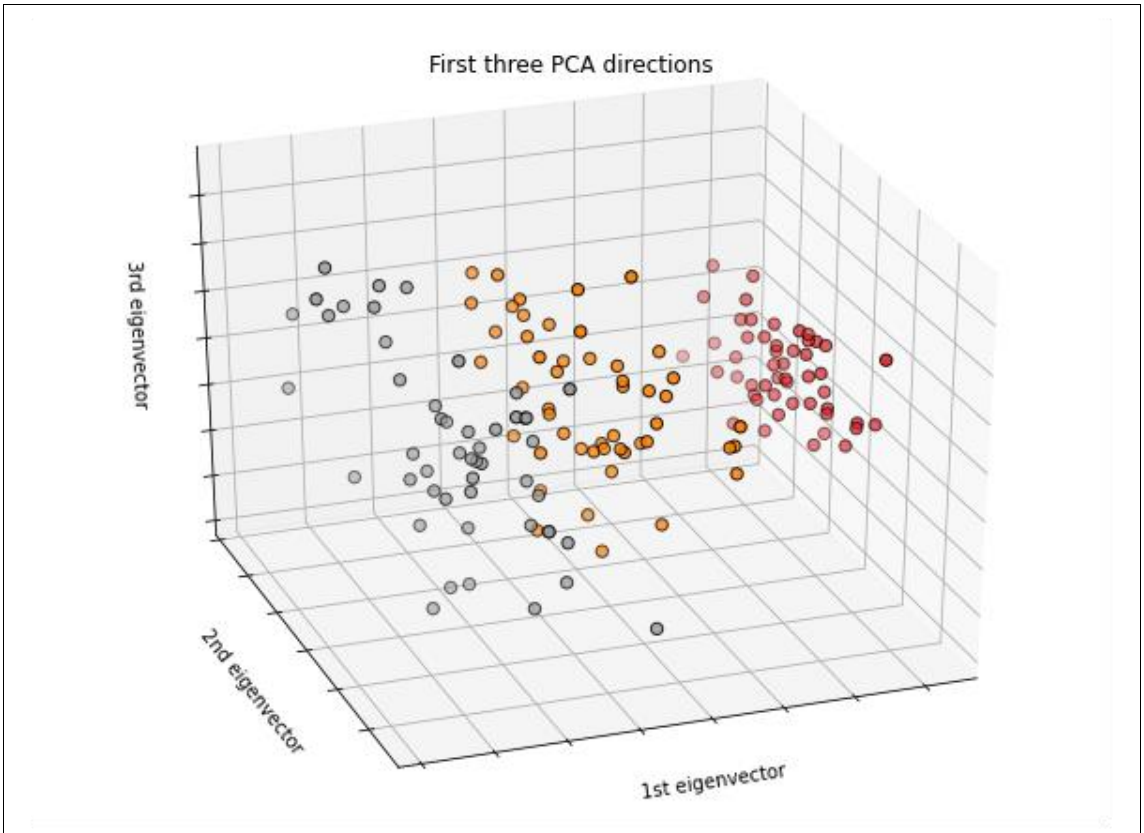

figure 3.2 -- 3D PCA

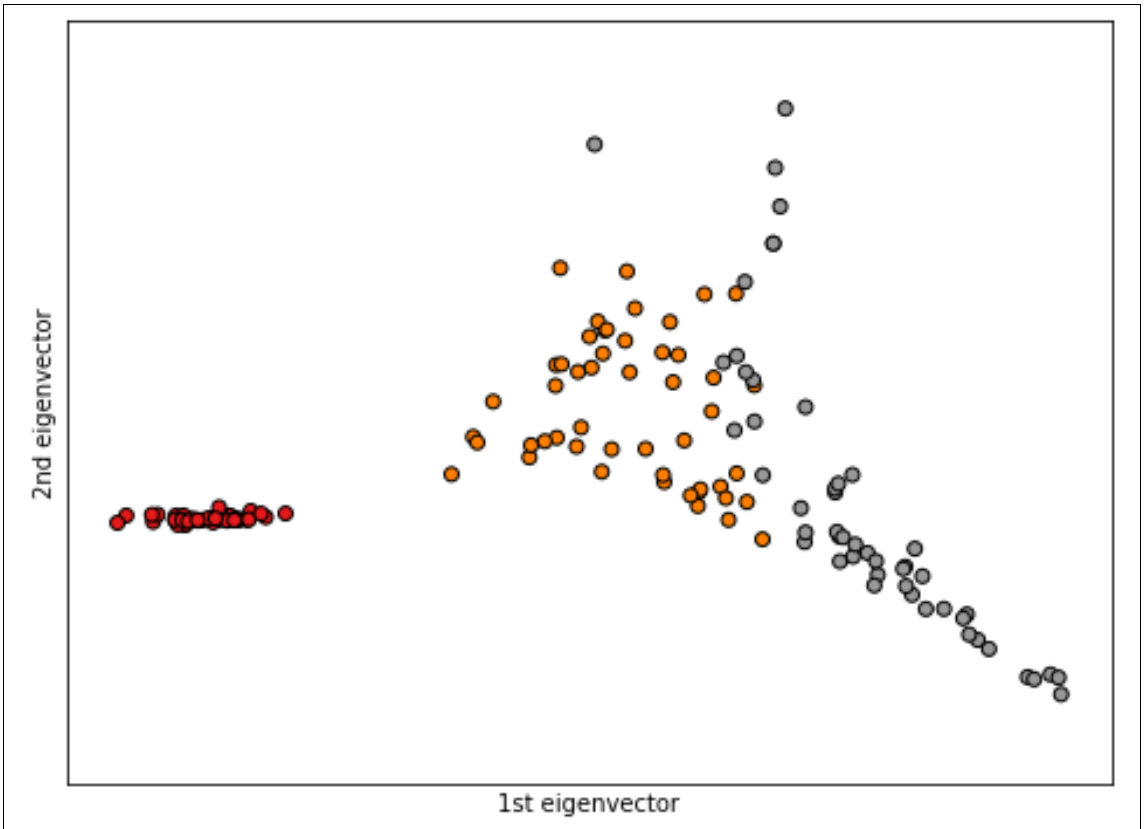

figure 3.3 -- 2D IsoMap

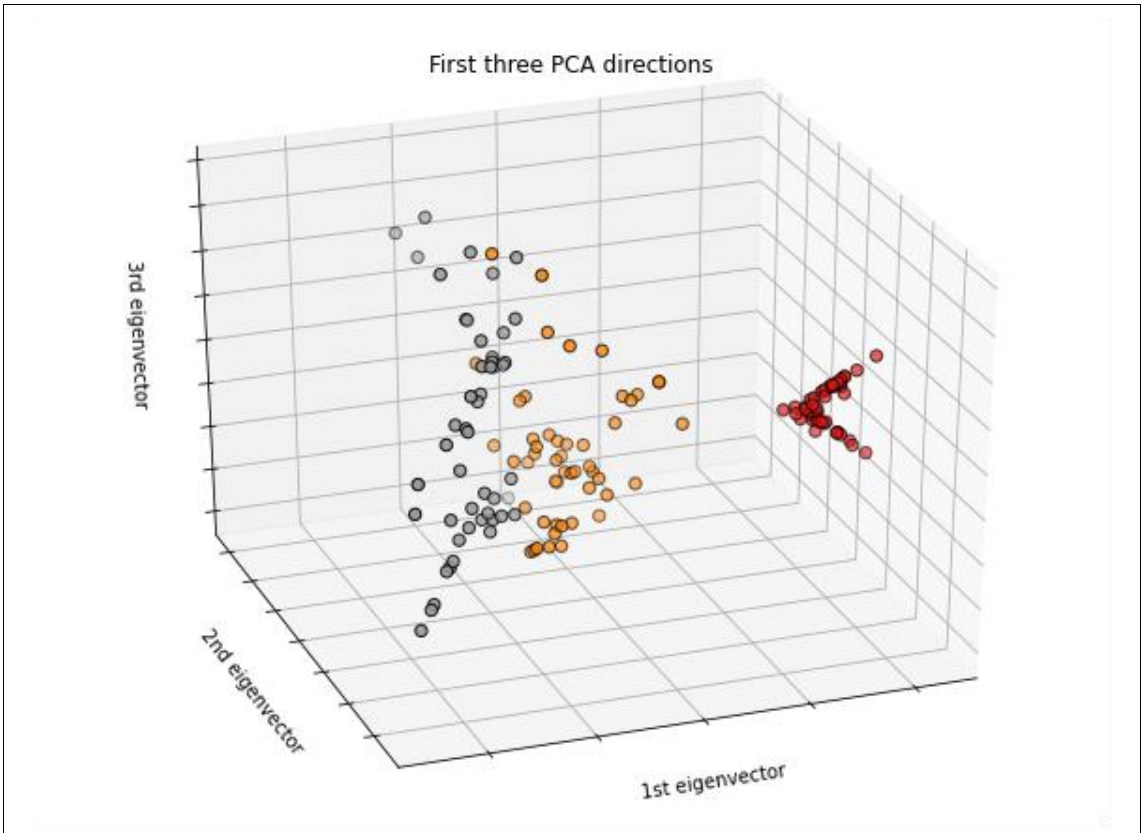

figure 3.4 -- 3D IsoMap

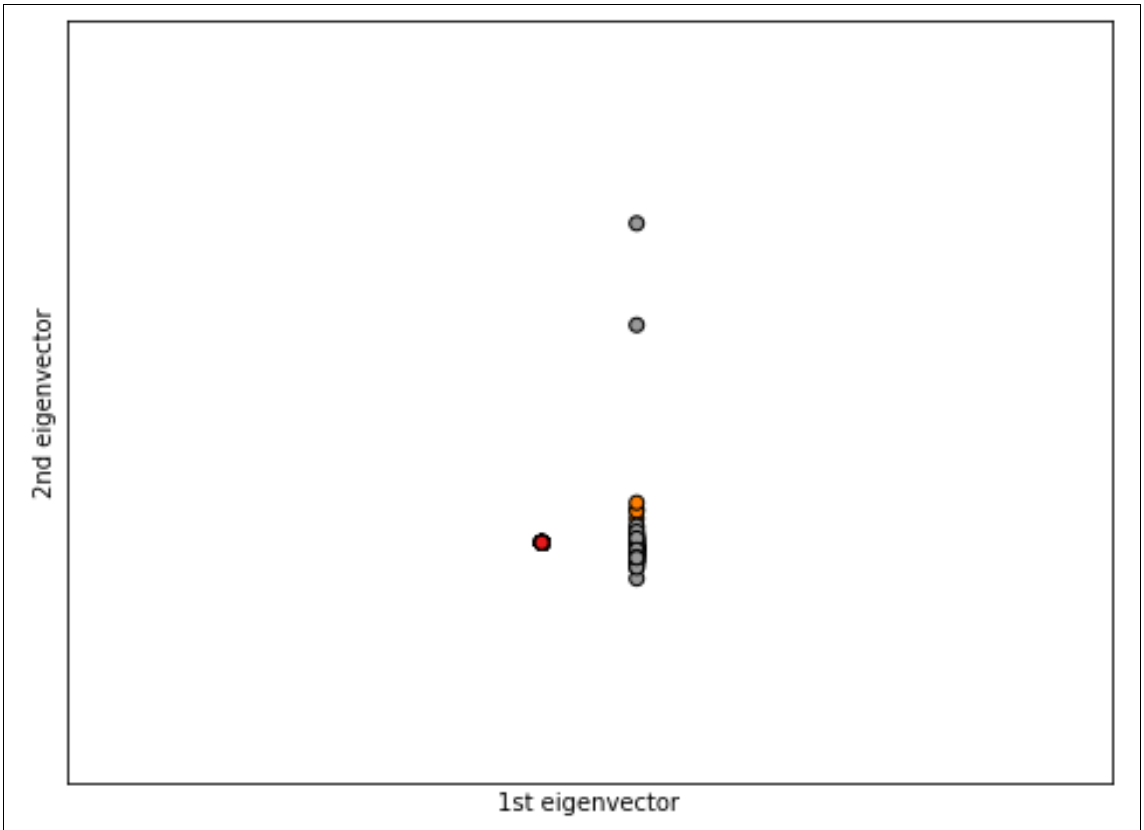

figure 3.5 -- 2D LLE

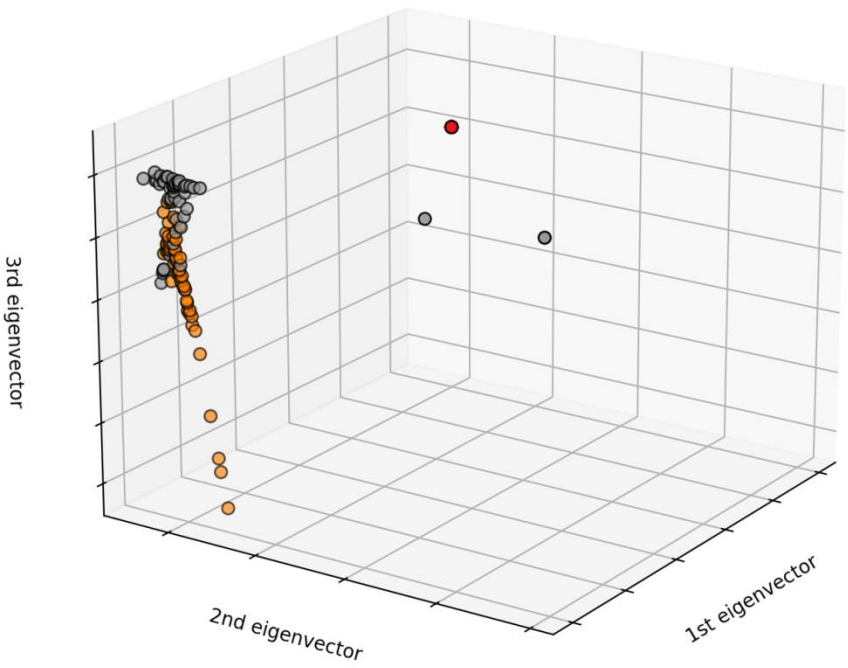

figure 3.6 -- 3D LLE

As you can see above, the PCA and IsoMap are relevant to the Iris dataset while LLE is hard to classify the different iris type.
In the PCA and IsoMap, you can easily classify the type of iris by human eye, but in the LLE you cannot distinguish the different types by eye.

-

Iillustrate in a table the best parameters that maximize the classification scores.

In the program, at the beginning, we use learning rate = 0.1, 10 epchos and momentum = 0.8 to run the sequantial model using keras. We notice that as the interation increase, the loss in decrease and score is increase.
Then, we change learning rate to lr * e $^{(-kt)}$ and 60 epochs. we got a smoother cruve but the loss value is higher than the previous one. Around 0.8 vs 0.35.
After that, we choose different optimizer and loss function. And we increase the accuracy to around 0.975.
Finally we use gridSearchCV and cross validation to tune the hyper parameters as fellow.

We have two table showing tuning parameters using cross validation.

Parameters

```
# define the grid search parameters
init_mode = ['uniform', 'lecun_uniform', 'normal', 'zero',
             'glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']
```

Result

```
Best Accuracy for 0.9700333476066589 using {'init_mode': 'he_normal'}
 mean=0.9652, std=0.002047 using {'init_mode': 'uniform'}
 mean=0.9674, std=0.0008981 using {'init_mode': 'lecun_uniform'}
 mean=0.9659, std=0.0009201 using {'init_mode': 'normal'}
 mean=0.1124, std=0.002416 using {'init_mode': 'zero'}
 mean=0.9687, std=0.0008841 using {'init_mode': 'glorot_normal'}
 mean=0.9681, std=0.001666 using {'init_mode': 'glorot_uniform'}
 mean=0.97, std=0.001555 using {'init_mode': 'he_normal'}
 mean=0.9691, std=0.001878 using {'init_mode': 'he_uniform'}
```

The first one is trying different weight of initialization. The highest score is he_normal

	init_mode	mean score	std
1	uniform	0.9652	0.002047
2	lecun_uniform	0.9674	0.0008981
3	normal	0.9659	0.0009201
4	zero	0.1124	0.002416
5	glorot_normal	0.9687	0.0008841
6	glorot_uniform	0.9681	0.001666
7	he_normal	0.97	0.001555
8	he_uniform	0.9691	0.001878

Parameters

```
init_mode = ['glorot_uniform', 'uniform']
batches = [128, 512]
epochs = [10, 20]
```

Result

```
Best Accuracy for 0.9721 using {'batch_size': 128, 'epochs': 20, 'init': 'glorot_uniform'}
mean=0.9673, std=0.001132 using {'batch_size': 128, 'epochs': 10, 'init': 'glorot_uniform'}
mean=0.9645, std=0.002879 using {'batch_size': 128, 'epochs': 10, 'init': 'uniform'}
mean=0.9721, std=0.001541 using {'batch_size': 128, 'epochs': 20, 'init': 'glorot_uniform'}
mean=0.9698, std=0.002253 using {'batch_size': 128, 'epochs': 20, 'init': 'uniform'}
mean=0.9581, std=0.002979 using {'batch_size': 512, 'epochs': 10, 'init': 'glorot_uniform'}
mean=0.9465, std=0.002678 using {'batch_size': 512, 'epochs': 10, 'init': 'uniform'}
mean=0.9682, std=0.0005099 using {'batch_size': 512, 'epochs': 20, 'init': 'glorot_uniform'}
mean=0.9637, std=0.001342 using {'batch_size': 512, 'epochs': 20, 'init': 'uniform'}
```

The second one is performing a GridSearch for batch size, number of epochs and initializer combined. The highest score is glorot_uniform with 128 bath_size and 20 phochs.

	init_mode	bath_size	epochs	mean score	std
1	glorot_uniform	128	10	0.9673	0.001132
2	uniform	128	10	0.9645	0.002879
3	glorot_uniform	128	20	0.9721	0.001514
4	uniform	128	20	0.9698	0.002253
5	glorot_uniform	512	10	0.9581	0.002979
6	uniform	512	10	0.9465	0.002678
7	glorot_uniform	512	20	0.9682	0.0005099
8	unfirom	512	20	0.9637	0.001342