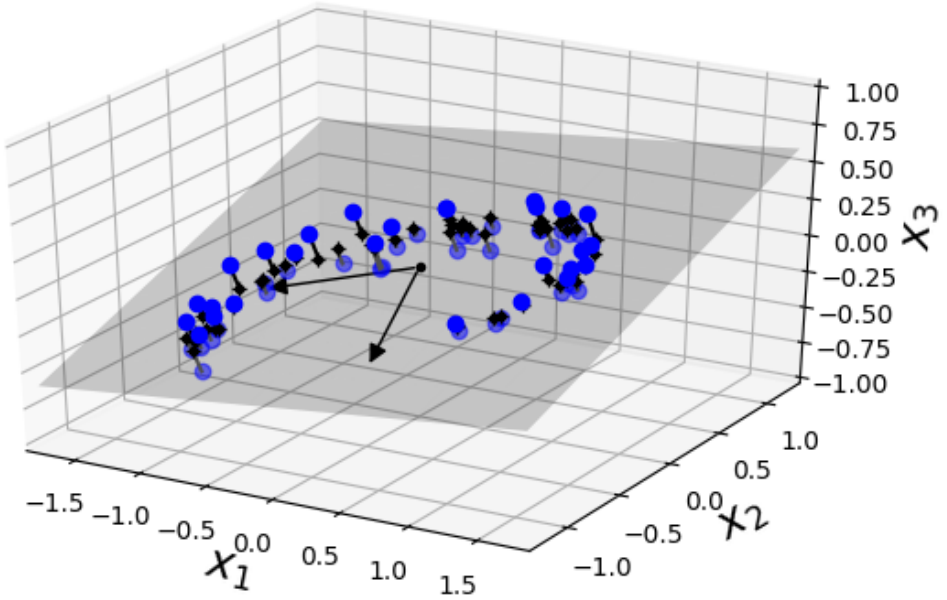
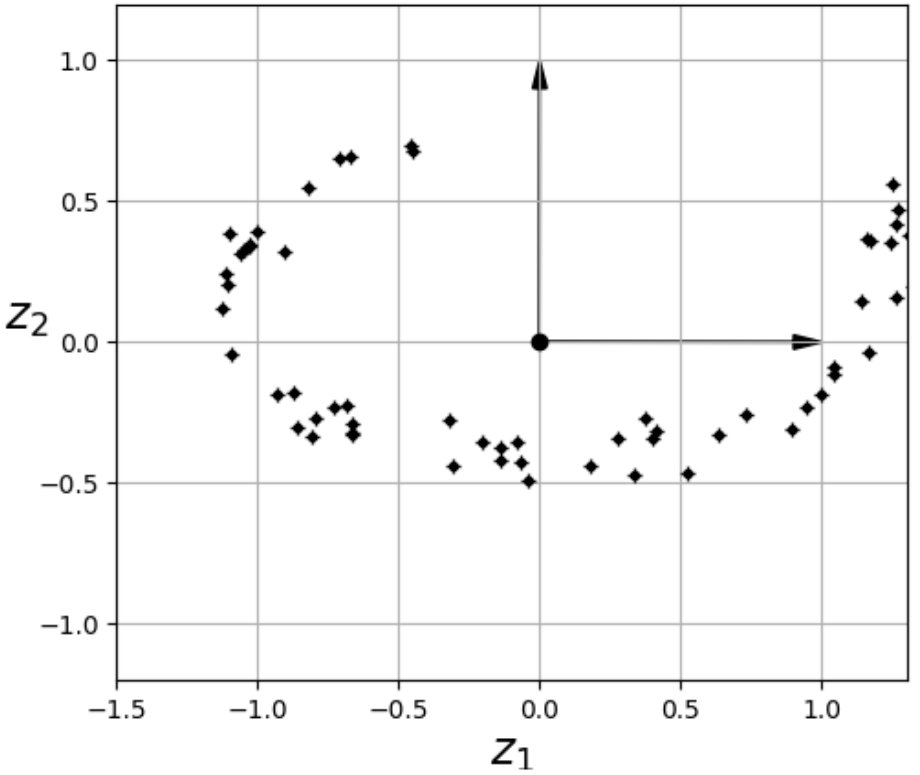
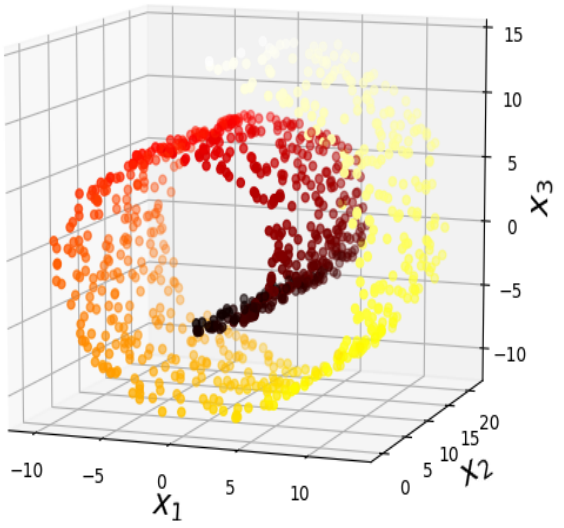
Dimensionality Reduction

Projection

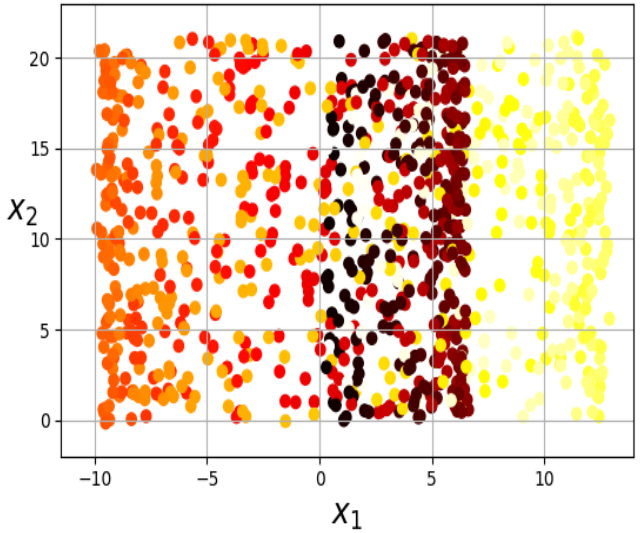

1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
x_3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
x_1
1.0
0.5
0.0
−0.5
−1.0
x_2

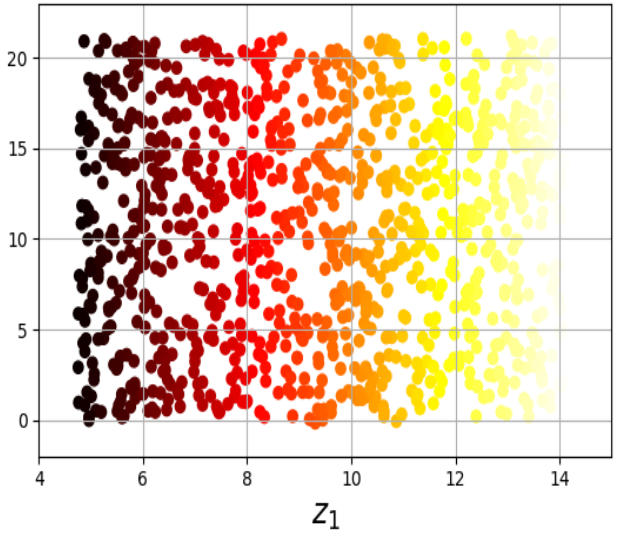

1.0
0.5
0.0
−0.5
−1.0
z_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
z_1

Projection

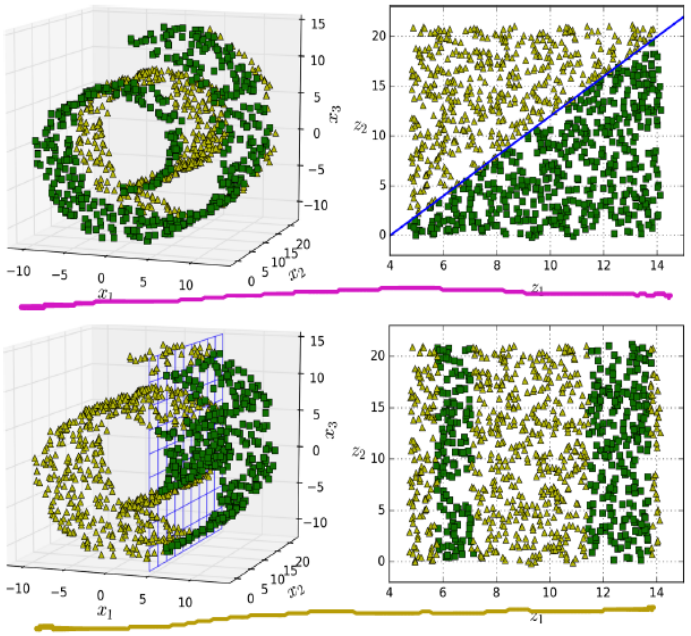


- Projection doesnot always do justice to the dataset.
- Unrolling would be a better option in this case.





Manifold

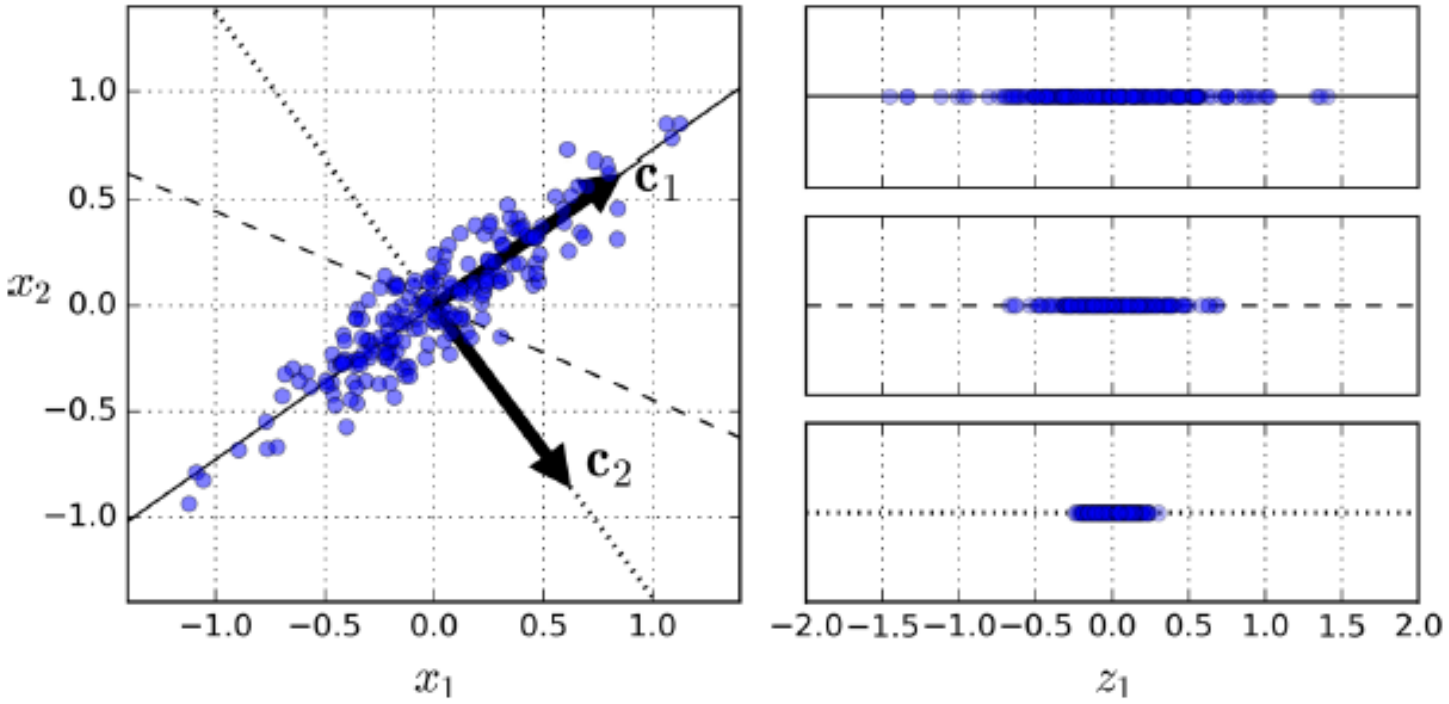


- DB is a straight in 2D space.
- DB around $x_1 = 5$ in 3D space, but not so in 2D space.

Manifold Learning

- Swiss roll is a 2D shape that can be bent into a higher dimension space.
- Generally d-dimension manifold is a part of n dimensional space ($d < n$)
- In reality most real world high-dimensional datasets lie close to a much lower dimensional manifold.
- Reducing dimensions will speed up training but may not always lead to better results.

PCA


1.0
0.5
0.0
−0.5
−1.0
x_2
−1.0 −0.5 0.0 0.5 1.0
x_1
$\mathbf{c}_1$
$\mathbf{c}_2$
−2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0
z_1

PCA

```
X, y = mnist["data"], mnist["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y)
X = X_train

pca = PCA()
pca.fit(X)
d = np.argmax(np.cumsum(pca.explained_variance_ratio_) >= 0.95) + 1
print("dimensions:",d)
```