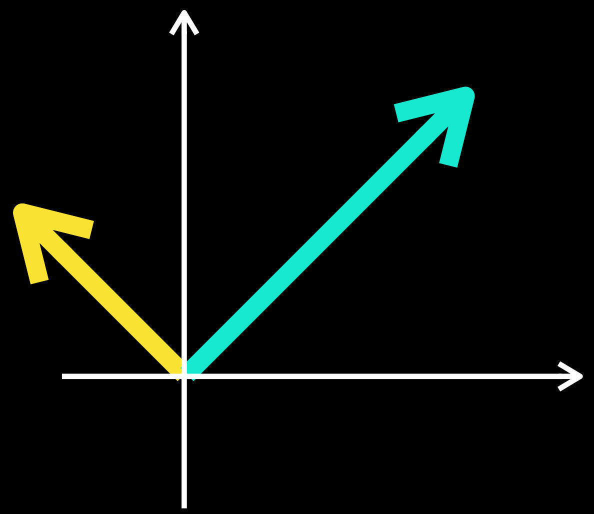
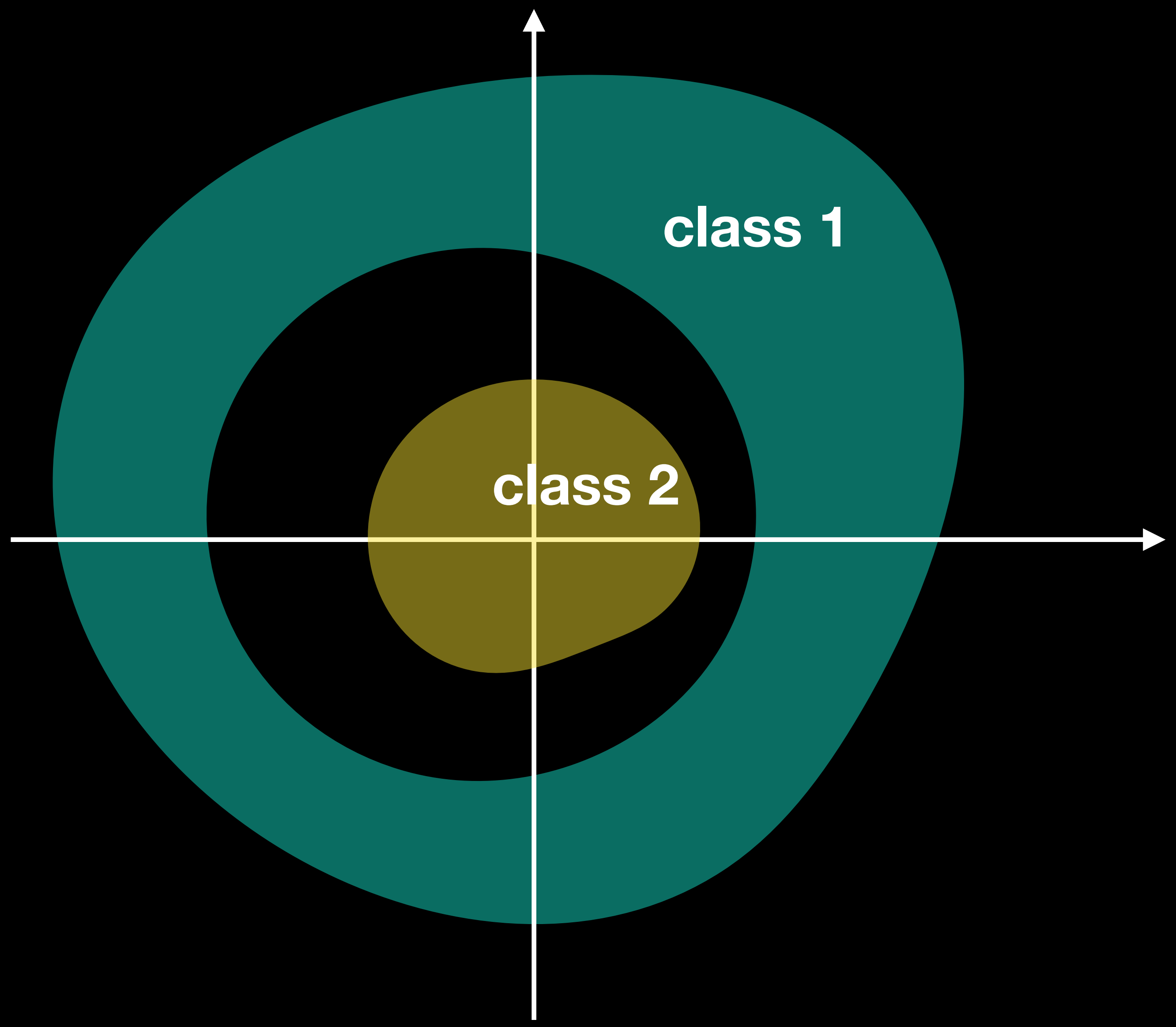
SVM Kernel Trick

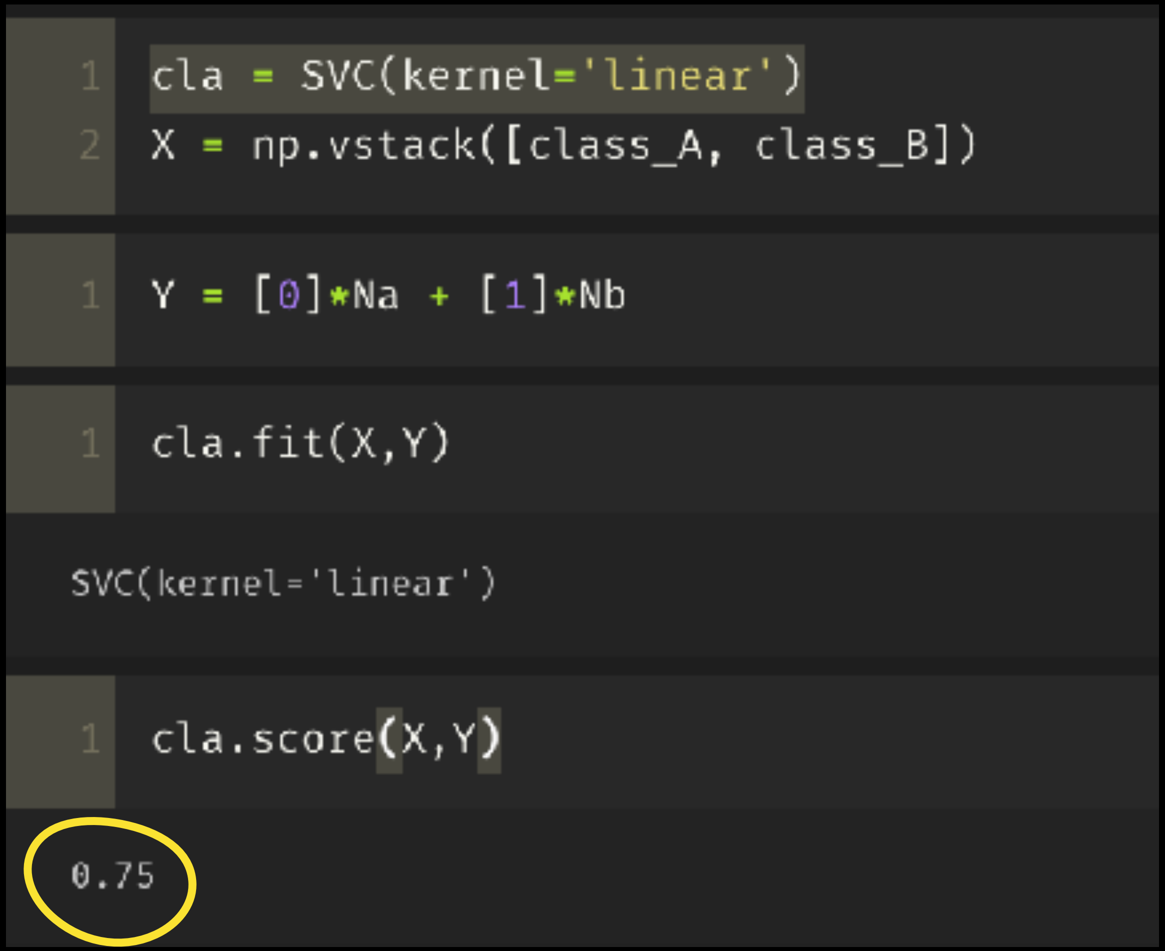
Linear Algebra Essentials



Nonlinear case



Example



```
cla = SVC(kernel='linear')
X = np.vstack([class_A, class_B])
```

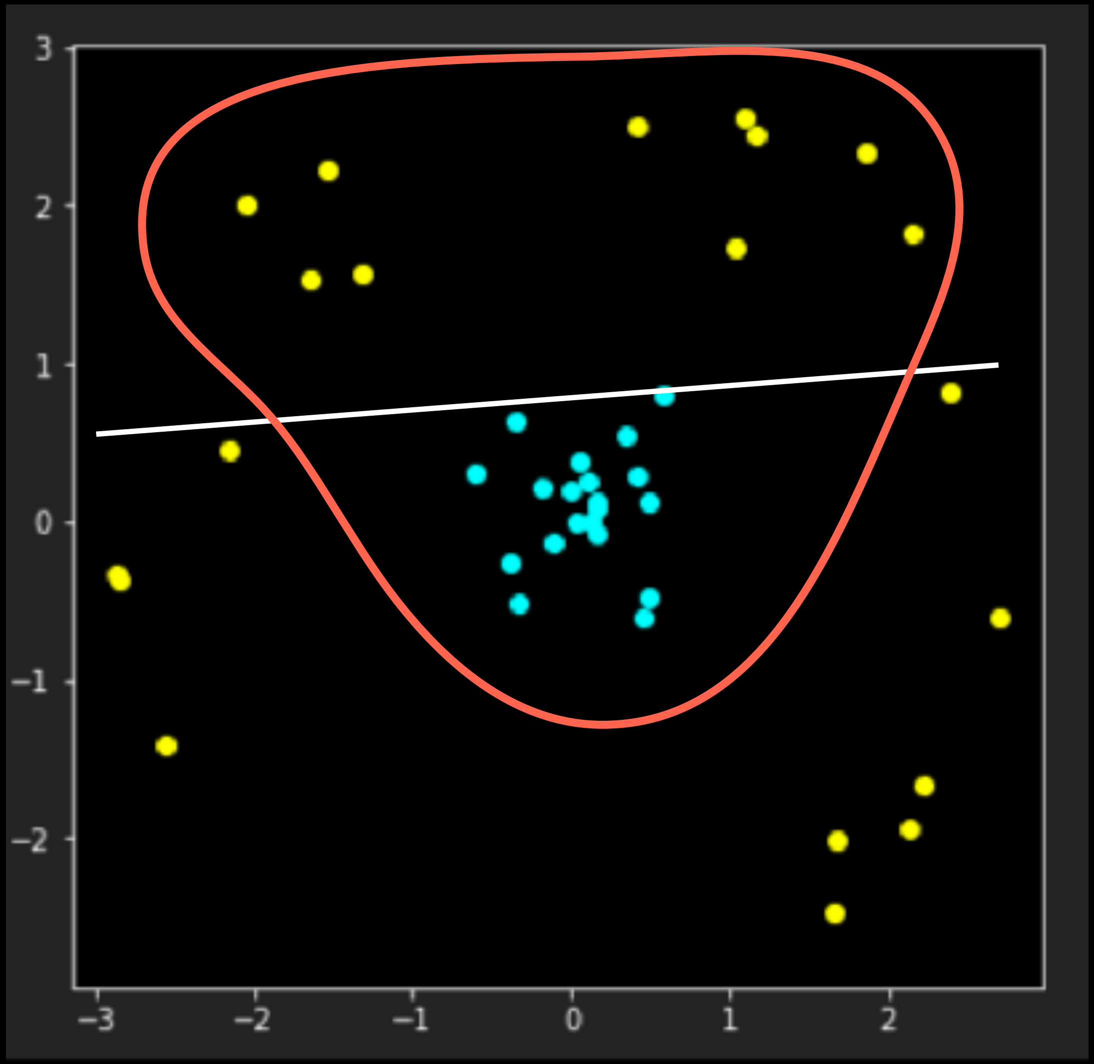
```
Y = [0]*Na + [1]*Nb
```

```
cla.fit(X,Y)
```

```
SVC(kernel='linear')
```

```
cla.score(X,Y)
```

```
0.75
```

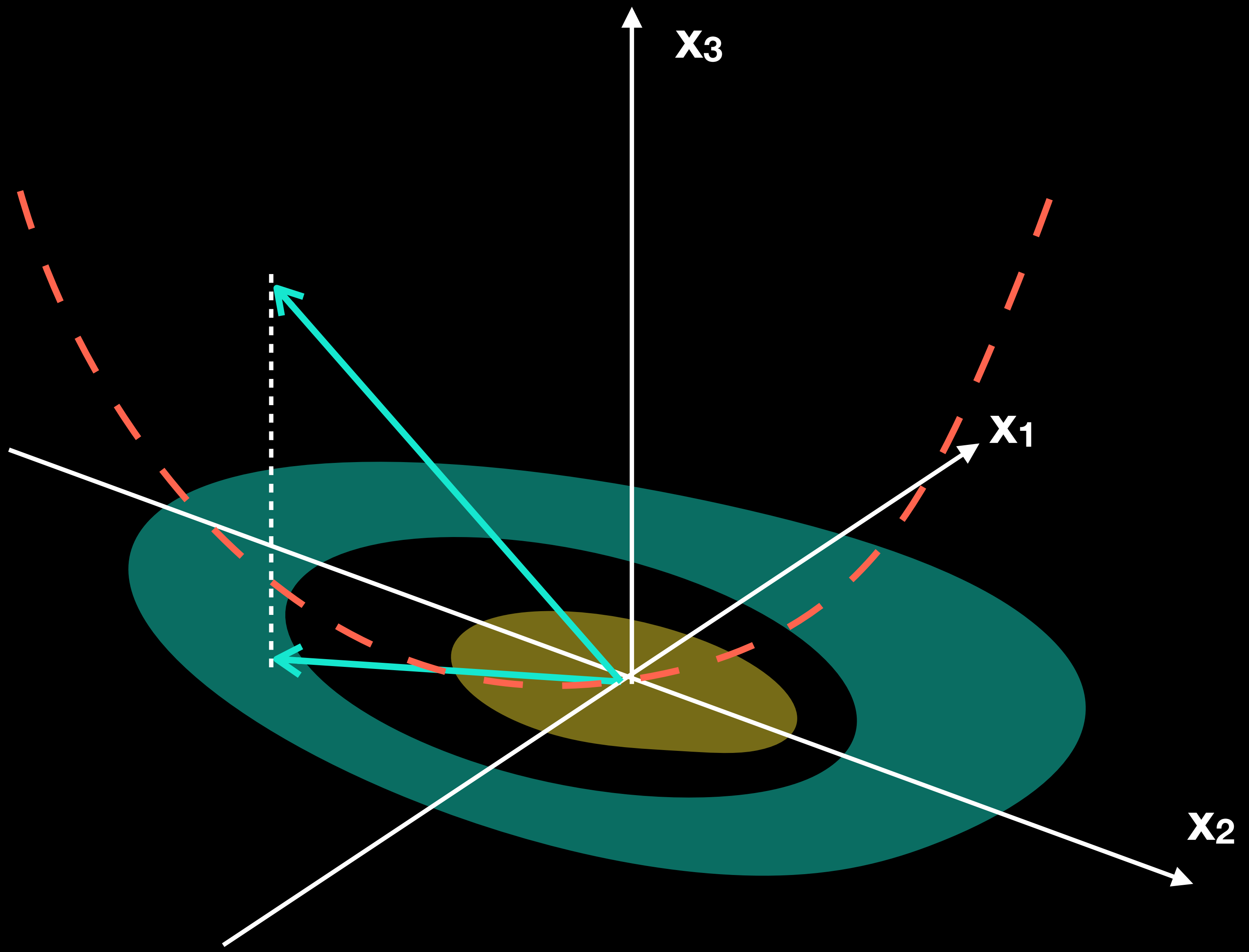


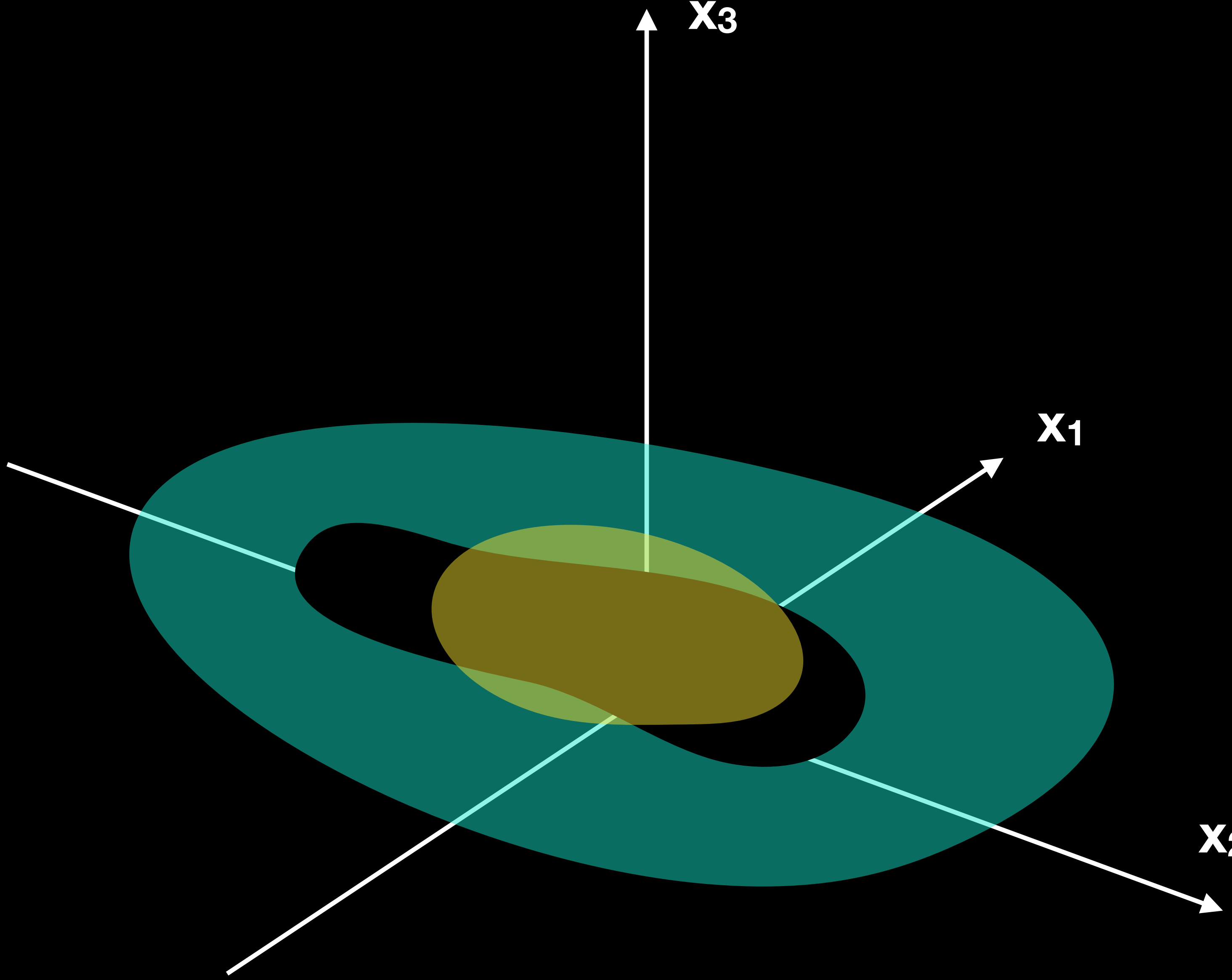
kernel trick

$$\overrightarrow{x} = [x_1, x_2]$$

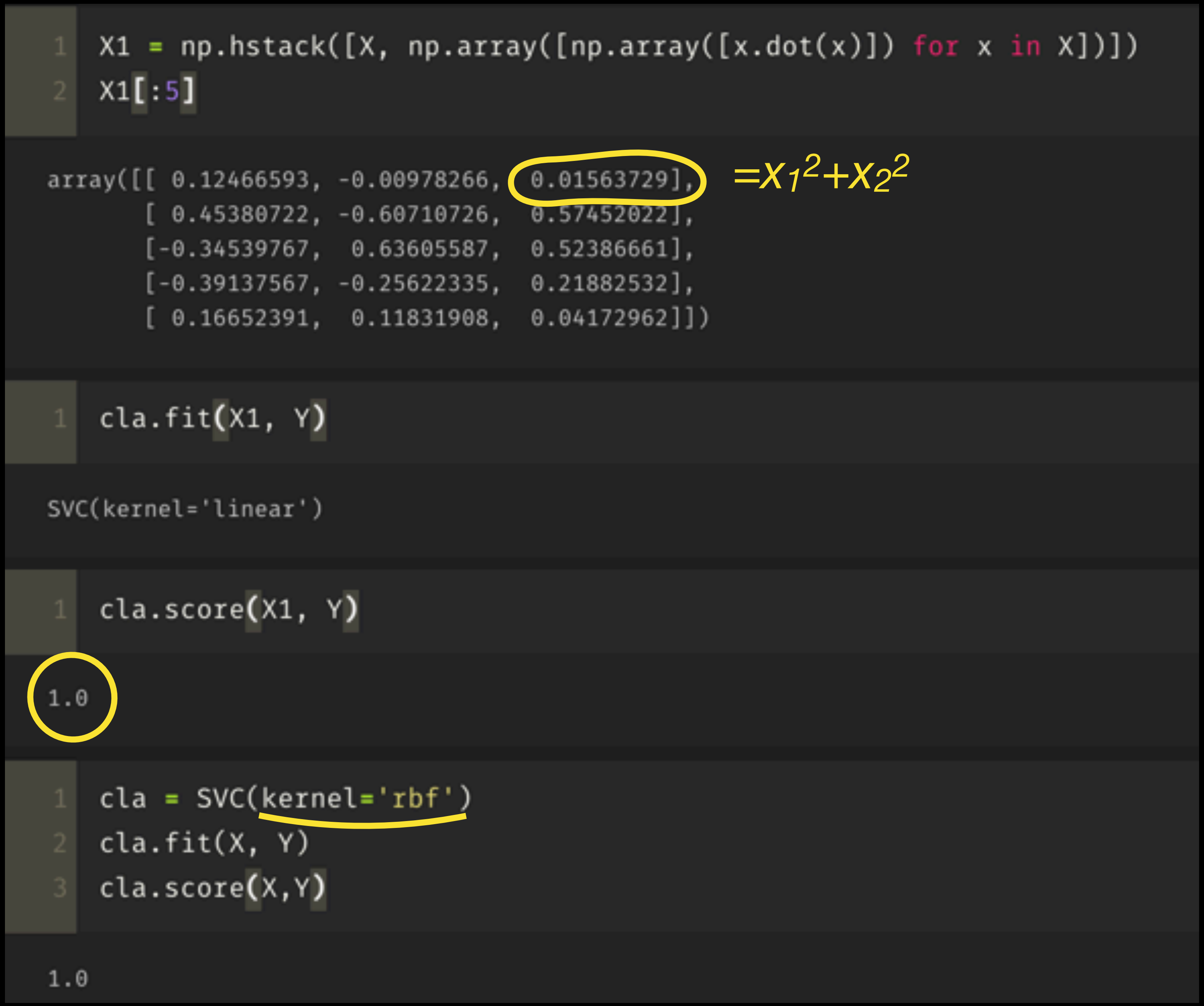
$$x_3 = \boxed{x_1^2 + x_2^2}$$

$$\overrightarrow{x}_{new} = [x_1, x_2, x_3]$$





Can be separated by hyperplane



```
X1 = np.hstack([X, np.array([np.array([x.dot(x)]) for x in X])])
X1[:5]
```

```
array([[ 0.12466593, -0.00978266,  0.01563729],
       [ 0.45380722, -0.60710726,  0.57452022],
       [-0.34539767,  0.63605587,  0.52386661],
       [-0.39137567, -0.25622335,  0.21882532],
       [ 0.16652391,  0.11831908,  0.04172962]])
```

$=x_1^2+x_2^2$

```
cla.fit(X1, Y)
```

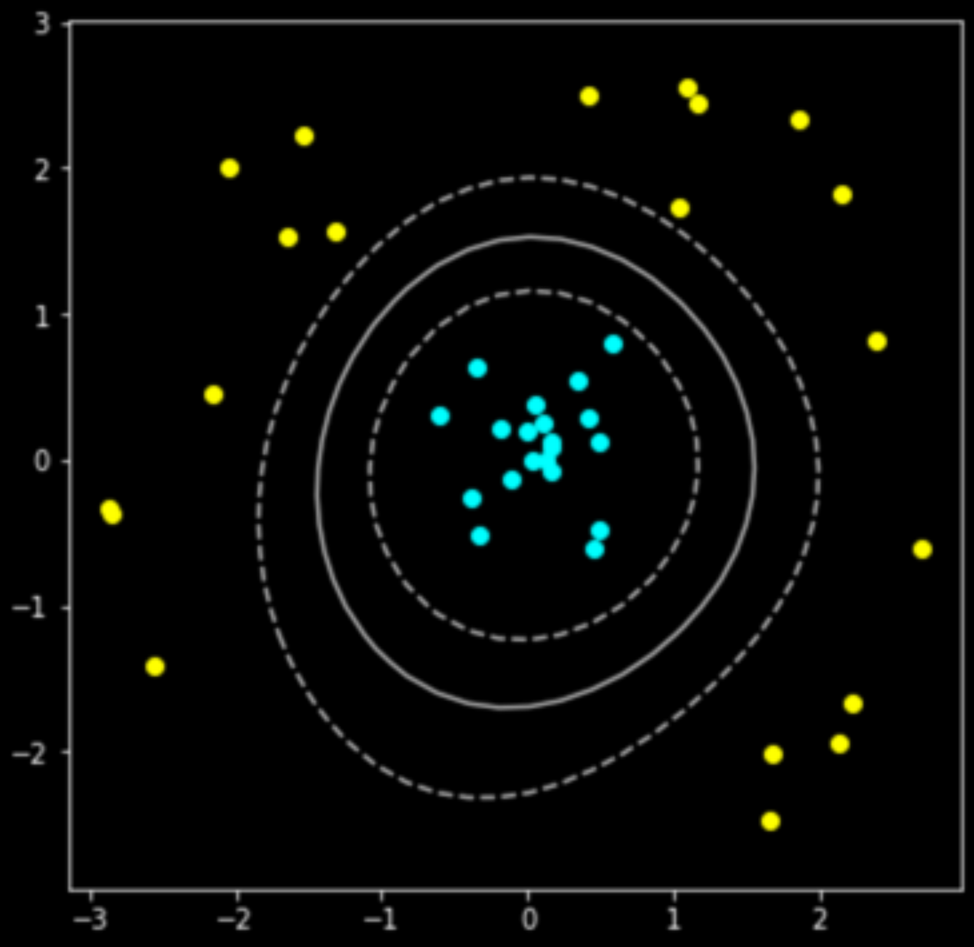
```
SVC(kernel='linear')
```

```
cla.score(X1, Y)
```

```
1.0
```

```
cla = SVC(kernel='rbf')
cla.fit(X, Y)
cla.score(X,Y)
```

```
1.0
```


3
2
1
0
−1
−2
−3
−2
−1
0
1
2