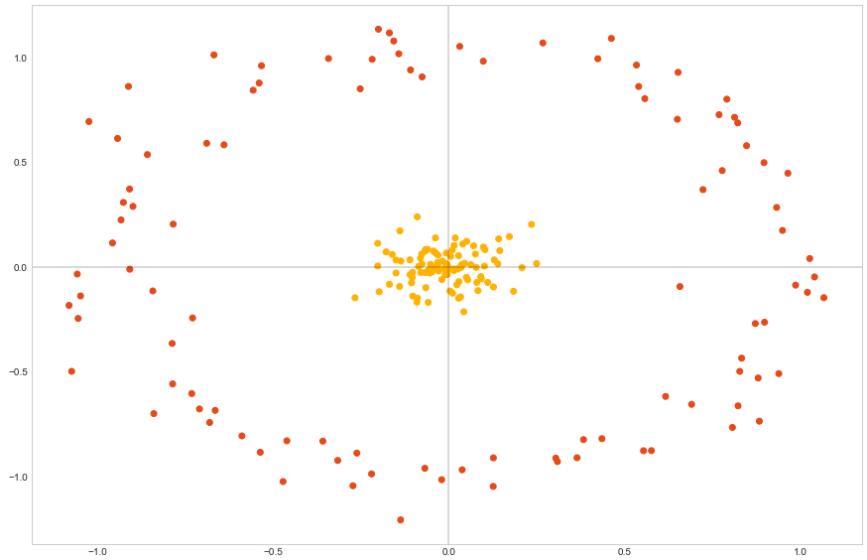
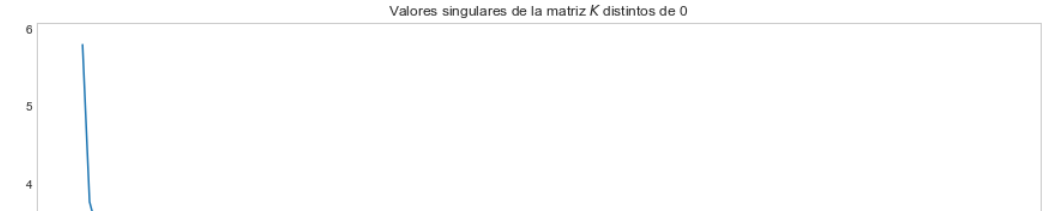
```
from sklearn.datasets import make_circles
from kpca import KPCA
from kernels import kernel
import matplotlib.pyplot as plt


X, Y = make_circles(n_samples=200, noise =0.1, factor= 0.0002)
colors = ['#FFB300' if e==1 else '#E64A19' for e in Y]


plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(15,10))
plt.axhline(c = 'black', alpha = 0.2)
plt.axvline(c = 'black', alpha = 0.2)
plt.scatter(X[:,0], X[:,1], c = colors)
plt.grid(False)
plt.show()
```



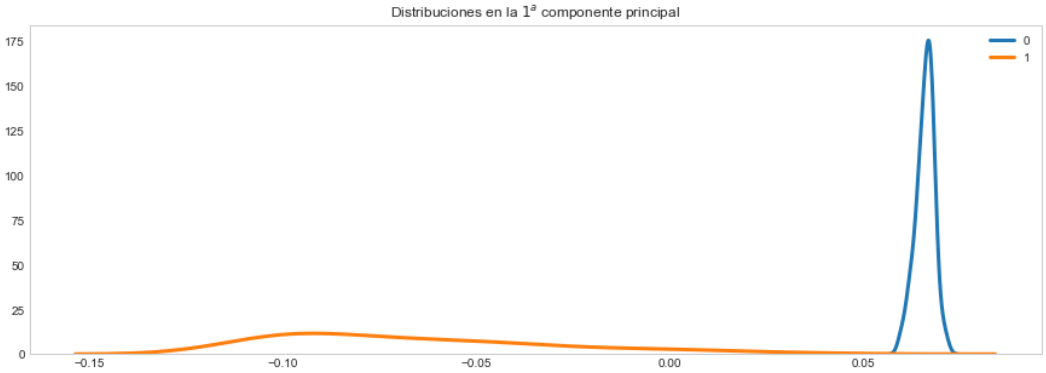

```
X = X.T # X must be dxn
k = kernel(gamma = 25).rbf
kpca = KPCA(X, k, 3)
scores = kpca.project()
```

```
C:\Users\Usuario\Desktop\Kernel-PCA\kpca.py:59: UserWarning: La matriz K no es semidefinida positiva
  warnings.warn("La matriz K no es semidefinida positiva")
```

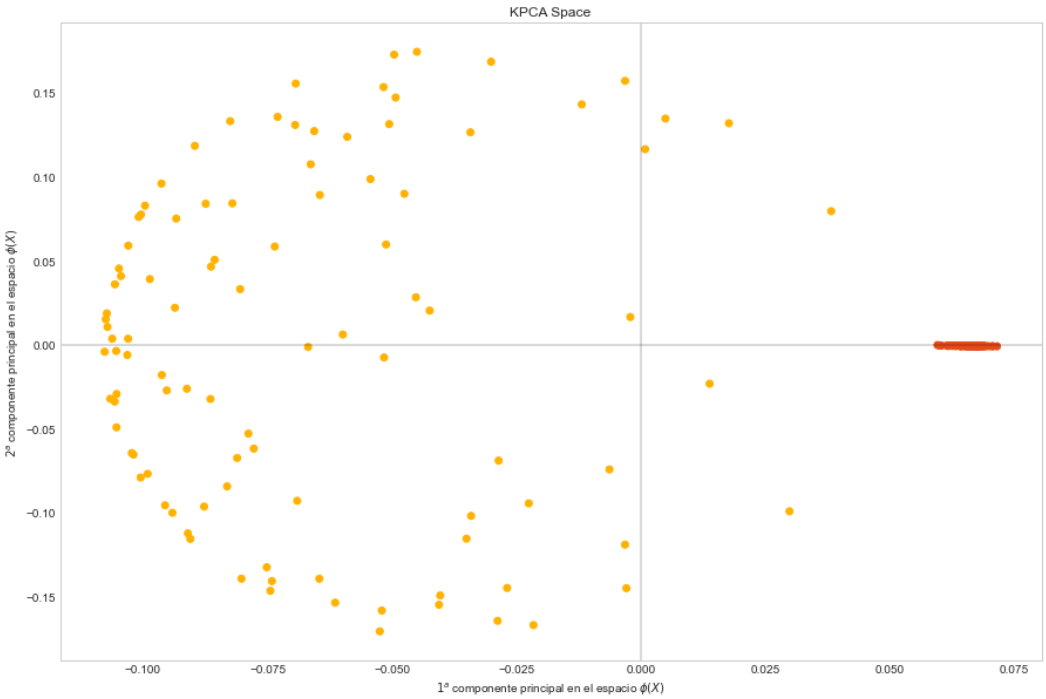
```
T = kpca.scores #Matrix of scores
K = kpca.K #Kernel matrix
V = kpca.v #Matrix of eigenvectors
S = kpca.sigma #Diagonal matrix of (real) singular values


kpca.plot_singular_values(grid = False)
```




```
kpca.plot_density(labels = Y, dim=1, grid = False)
```




```
kpca.plot_scores_2d(colors = colors, grid = False, dim_1 = 1, dim_2 = 2)
```




```
kpca.plot_scores_3d(colors = colors, dim_1 = 1, dim_2 = 2, dim_3 = 3)
```