# trab4

November 28, 2020

## 1 Neste trabalho (Trabalho 4), a idéia é ter uma revisão de PCA, passando por seus conceitos e construindo um pouco de código

É essencial ver os códigos e comentários de cada célula, além das 3 questões...

[4]:
```
import numpy as np
import pandas as pd
from sklearn import linear_model
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
```

[5]:
```
df=pd.read_csv('iris.txt',names=['m1','m2','m3','m4','esp']) #acertar path para␣
 ↪o dataset
```

[9]:
```
df.head()
```

[9]:

| | m1 | m2 | m3 | m4 | esp |
|---|---|---|---|---|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

Passo 1: Criar uma matriz com as features numéricas

Standardizar as Features: importante para PCA, KNN,K-means...na dúvida, standardizar!

[22]:
```
X=df.to_numpy()[ : , :4]
scaler=StandardScaler()
scaler.fit(X)
XS=scaler.transform(X)
XS[:]
```

[22]:
```
array([[-9.00681170e-01,  1.03205722e+00, -1.34127240e+00,
        -1.31297673e+00],
       [-1.14301691e+00, -1.24957601e-01, -1.34127240e+00,
        -1.31297673e+00],
```

```
[-1.38535265e+00,  3.37848329e-01, -1.39813811e+00,
 -1.31297673e+00],
[-1.50652052e+00,  1.06445364e-01, -1.28440670e+00,
 -1.31297673e+00],
[-1.02184904e+00,  1.26346019e+00, -1.34127240e+00,
 -1.31297673e+00],
[-5.37177559e-01,  1.95766909e+00, -1.17067529e+00,
 -1.05003079e+00],
[-1.50652052e+00,  8.00654259e-01, -1.34127240e+00,
 -1.18150376e+00],
[-1.02184904e+00,  8.00654259e-01, -1.28440670e+00,
 -1.31297673e+00],
[-1.74885626e+00, -3.56360566e-01, -1.34127240e+00,
 -1.31297673e+00],
[-1.14301691e+00,  1.06445364e-01, -1.28440670e+00,
 -1.44444970e+00],
[-5.37177559e-01,  1.49486315e+00, -1.28440670e+00,
 -1.31297673e+00],
[-1.26418478e+00,  8.00654259e-01, -1.22754100e+00,
 -1.31297673e+00],
[-1.26418478e+00, -1.24957601e-01, -1.34127240e+00,
 -1.44444970e+00],
[-1.87002413e+00, -1.24957601e-01, -1.51186952e+00,
 -1.44444970e+00],
[-5.25060772e-02,  2.18907205e+00, -1.45500381e+00,
 -1.31297673e+00],
[-1.73673948e-01,  3.11468391e+00, -1.28440670e+00,
 -1.05003079e+00],
[-5.37177559e-01,  1.95766909e+00, -1.39813811e+00,
 -1.05003079e+00],
[-9.00681170e-01,  1.03205722e+00, -1.34127240e+00,
 -1.18150376e+00],
[-1.73673948e-01,  1.72626612e+00, -1.17067529e+00,
 -1.18150376e+00],
[-9.00681170e-01,  1.72626612e+00, -1.28440670e+00,
 -1.18150376e+00],
[-5.37177559e-01,  8.00654259e-01, -1.17067529e+00,
 -1.31297673e+00],
[-9.00681170e-01,  1.49486315e+00, -1.28440670e+00,
 -1.05003079e+00],
[-1.50652052e+00,  1.26346019e+00, -1.56873522e+00,
 -1.31297673e+00],
[-9.00681170e-01,  5.69251294e-01, -1.17067529e+00,
 -9.18557817e-01],
[-1.26418478e+00,  8.00654259e-01, -1.05694388e+00,
 -1.31297673e+00],
[-1.02184904e+00, -1.24957601e-01, -1.22754100e+00,
```

```
 -1.31297673e+00],
[-1.02184904e+00,  8.00654259e-01, -1.22754100e+00,
 -1.05003079e+00],
[-7.79513300e-01,  1.03205722e+00, -1.28440670e+00,
 -1.31297673e+00],
[-7.79513300e-01,  8.00654259e-01, -1.34127240e+00,
 -1.31297673e+00],
[-1.38535265e+00,  3.37848329e-01, -1.22754100e+00,
 -1.31297673e+00],
[-1.26418478e+00,  1.06445364e-01, -1.22754100e+00,
 -1.31297673e+00],
[-5.37177559e-01,  8.00654259e-01, -1.28440670e+00,
 -1.05003079e+00],
[-7.79513300e-01,  2.42047502e+00, -1.28440670e+00,
 -1.44444970e+00],
[-4.16009689e-01,  2.65187798e+00, -1.34127240e+00,
 -1.31297673e+00],
[-1.14301691e+00,  1.06445364e-01, -1.28440670e+00,
 -1.44444970e+00],
[-1.02184904e+00,  3.37848329e-01, -1.45500381e+00,
 -1.31297673e+00],
[-4.16009689e-01,  1.03205722e+00, -1.39813811e+00,
 -1.31297673e+00],
[-1.14301691e+00,  1.06445364e-01, -1.28440670e+00,
 -1.44444970e+00],
[-1.74885626e+00, -1.24957601e-01, -1.39813811e+00,
 -1.31297673e+00],
[-9.00681170e-01,  8.00654259e-01, -1.28440670e+00,
 -1.31297673e+00],
[-1.02184904e+00,  1.03205722e+00, -1.39813811e+00,
 -1.18150376e+00],
[-1.62768839e+00, -1.74477836e+00, -1.39813811e+00,
 -1.18150376e+00],
[-1.74885626e+00,  3.37848329e-01, -1.39813811e+00,
 -1.31297673e+00],
[-1.02184904e+00,  1.03205722e+00, -1.22754100e+00,
 -7.87084847e-01],
[-9.00681170e-01,  1.72626612e+00, -1.05694388e+00,
 -1.05003079e+00],
[-1.26418478e+00, -1.24957601e-01, -1.34127240e+00,
 -1.18150376e+00],
[-9.00681170e-01,  1.72626612e+00, -1.22754100e+00,
 -1.31297673e+00],
[-1.50652052e+00,  3.37848329e-01, -1.34127240e+00,
 -1.31297673e+00],
[-6.58345429e-01,  1.49486315e+00, -1.28440670e+00,
 -1.31297673e+00],
```

```
[-1.02184904e+00,  5.69251294e-01, -1.34127240e+00,
 -1.31297673e+00],
[ 1.40150837e+00,  3.37848329e-01,  5.35295827e-01,
  2.64698913e-01],
[ 6.74501145e-01,  3.37848329e-01,  4.21564419e-01,
  3.96171883e-01],
[ 1.28034050e+00,  1.06445364e-01,  6.49027235e-01,
  3.96171883e-01],
[-4.16009689e-01, -1.74477836e+00,  1.37235899e-01,
  1.33225943e-01],
[ 7.95669016e-01, -5.87763531e-01,  4.78430123e-01,
  3.96171883e-01],
[-1.73673948e-01, -5.87763531e-01,  4.21564419e-01,
  1.33225943e-01],
[ 5.53333275e-01,  5.69251294e-01,  5.35295827e-01,
  5.27644853e-01],
[-1.14301691e+00, -1.51337539e+00, -2.60824029e-01,
 -2.61192967e-01],
[ 9.16836886e-01, -3.56360566e-01,  4.78430123e-01,
  1.33225943e-01],
[-7.79513300e-01, -8.19166497e-01,  8.03701950e-02,
  2.64698913e-01],
[-1.02184904e+00, -2.43898725e+00, -1.47092621e-01,
 -2.61192967e-01],
[ 6.86617933e-02, -1.24957601e-01,  2.50967307e-01,
  3.96171883e-01],
[ 1.89829664e-01, -1.97618132e+00,  1.37235899e-01,
 -2.61192967e-01],
[ 3.10997534e-01, -3.56360566e-01,  5.35295827e-01,
  2.64698913e-01],
[-2.94841818e-01, -3.56360566e-01, -9.02269170e-02,
  1.33225943e-01],
[ 1.03800476e+00,  1.06445364e-01,  3.64698715e-01,
  2.64698913e-01],
[-2.94841818e-01, -1.24957601e-01,  4.21564419e-01,
  3.96171883e-01],
[-5.25060772e-02, -8.19166497e-01,  1.94101603e-01,
 -2.61192967e-01],
[ 4.32165405e-01, -1.97618132e+00,  4.21564419e-01,
  3.96171883e-01],
[-2.94841818e-01, -1.28197243e+00,  8.03701950e-02,
 -1.29719997e-01],
[ 6.86617933e-02,  3.37848329e-01,  5.92161531e-01,
  7.90590793e-01],
[ 3.10997534e-01, -5.87763531e-01,  1.37235899e-01,
  1.33225943e-01],
[ 5.53333275e-01, -1.28197243e+00,  6.49027235e-01,
```

```
  3.96171883e-01],
[ 3.10997534e-01, -5.87763531e-01,  5.35295827e-01,
  1.75297293e-03],
[ 6.74501145e-01, -3.56360566e-01,  3.07833011e-01,
  1.33225943e-01],
[ 9.16836886e-01, -1.24957601e-01,  3.64698715e-01,
  2.64698913e-01],
[ 1.15917263e+00, -5.87763531e-01,  5.92161531e-01,
  2.64698913e-01],
[ 1.03800476e+00, -1.24957601e-01,  7.05892939e-01,
  6.59117823e-01],
[ 1.89829664e-01, -3.56360566e-01,  4.21564419e-01,
  3.96171883e-01],
[-1.73673948e-01, -1.05056946e+00, -1.47092621e-01,
 -2.61192967e-01],
[-4.16009689e-01, -1.51337539e+00,  2.35044910e-02,
 -1.29719997e-01],
[-4.16009689e-01, -1.51337539e+00, -3.33612130e-02,
 -2.61192967e-01],
[-5.25060772e-02, -8.19166497e-01,  8.03701950e-02,
  1.75297293e-03],
[ 1.89829664e-01, -8.19166497e-01,  7.62758643e-01,
  5.27644853e-01],
[-5.37177559e-01, -1.24957601e-01,  4.21564419e-01,
  3.96171883e-01],
[ 1.89829664e-01,  8.00654259e-01,  4.21564419e-01,
  5.27644853e-01],
[ 1.03800476e+00,  1.06445364e-01,  5.35295827e-01,
  3.96171883e-01],
[ 5.53333275e-01, -1.74477836e+00,  3.64698715e-01,
  1.33225943e-01],
[-2.94841818e-01, -1.24957601e-01,  1.94101603e-01,
  1.33225943e-01],
[-4.16009689e-01, -1.28197243e+00,  1.37235899e-01,
  1.33225943e-01],
[-4.16009689e-01, -1.05056946e+00,  3.64698715e-01,
  1.75297293e-03],
[ 3.10997534e-01, -1.24957601e-01,  4.78430123e-01,
  2.64698913e-01],
[-5.25060772e-02, -1.05056946e+00,  1.37235899e-01,
  1.75297293e-03],
[-1.02184904e+00, -1.74477836e+00, -2.60824029e-01,
 -2.61192967e-01],
[-2.94841818e-01, -8.19166497e-01,  2.50967307e-01,
  1.33225943e-01],
[-1.73673948e-01, -1.24957601e-01,  2.50967307e-01,
  1.75297293e-03],
```

```
[-1.73673948e-01, -3.56360566e-01,  2.50967307e-01,
  1.33225943e-01],
[ 4.32165405e-01, -3.56360566e-01,  3.07833011e-01,
  1.33225943e-01],
[-9.00681170e-01, -1.28197243e+00, -4.31421141e-01,
 -1.29719997e-01],
[-1.73673948e-01, -5.87763531e-01,  1.94101603e-01,
  1.33225943e-01],
[ 5.53333275e-01,  5.69251294e-01,  1.27454998e+00,
  1.71090158e+00],
[-5.25060772e-02, -8.19166497e-01,  7.62758643e-01,
  9.22063763e-01],
[ 1.52267624e+00, -1.24957601e-01,  1.21768427e+00,
  1.18500970e+00],
[ 5.53333275e-01, -3.56360566e-01,  1.04708716e+00,
  7.90590793e-01],
[ 7.95669016e-01, -1.24957601e-01,  1.16081857e+00,
  1.31648267e+00],
[ 2.12851559e+00, -1.24957601e-01,  1.61574420e+00,
  1.18500970e+00],
[-1.14301691e+00, -1.28197243e+00,  4.21564419e-01,
  6.59117823e-01],
[ 1.76501198e+00, -3.56360566e-01,  1.44514709e+00,
  7.90590793e-01],
[ 1.03800476e+00, -1.28197243e+00,  1.16081857e+00,
  7.90590793e-01],
[ 1.64384411e+00,  1.26346019e+00,  1.33141568e+00,
  1.71090158e+00],
[ 7.95669016e-01,  3.37848329e-01,  7.62758643e-01,
  1.05353673e+00],
[ 6.74501145e-01, -8.19166497e-01,  8.76490051e-01,
  9.22063763e-01],
[ 1.15917263e+00, -1.24957601e-01,  9.90221459e-01,
  1.18500970e+00],
[-1.73673948e-01, -1.28197243e+00,  7.05892939e-01,
  1.05353673e+00],
[-5.25060772e-02, -5.87763531e-01,  7.62758643e-01,
  1.57942861e+00],
[ 6.74501145e-01,  3.37848329e-01,  8.76490051e-01,
  1.44795564e+00],
[ 7.95669016e-01, -1.24957601e-01,  9.90221459e-01,
  7.90590793e-01],
[ 2.24968346e+00,  1.72626612e+00,  1.67260991e+00,
  1.31648267e+00],
[ 2.24968346e+00, -1.05056946e+00,  1.78634131e+00,
  1.44795564e+00],
[ 1.89829664e-01, -1.97618132e+00,  7.05892939e-01,
```

```
  3.96171883e-01],
[ 1.28034050e+00,  3.37848329e-01,  1.10395287e+00,
  1.44795564e+00],
[-2.94841818e-01, -5.87763531e-01,  6.49027235e-01,
  1.05353673e+00],
[ 2.24968346e+00, -5.87763531e-01,  1.67260991e+00,
  1.05353673e+00],
[ 5.53333275e-01, -8.19166497e-01,  6.49027235e-01,
  7.90590793e-01],
[ 1.03800476e+00,  5.69251294e-01,  1.10395287e+00,
  1.18500970e+00],
[ 1.64384411e+00,  3.37848329e-01,  1.27454998e+00,
  7.90590793e-01],
[ 4.32165405e-01, -5.87763531e-01,  5.92161531e-01,
  7.90590793e-01],
[ 3.10997534e-01, -1.24957601e-01,  6.49027235e-01,
  7.90590793e-01],
[ 6.74501145e-01, -5.87763531e-01,  1.04708716e+00,
  1.18500970e+00],
[ 1.64384411e+00, -1.24957601e-01,  1.16081857e+00,
  5.27644853e-01],
[ 1.88617985e+00, -5.87763531e-01,  1.33141568e+00,
  9.22063763e-01],
[ 2.49201920e+00,  1.72626612e+00,  1.50201279e+00,
  1.05353673e+00],
[ 6.74501145e-01, -5.87763531e-01,  1.04708716e+00,
  1.31648267e+00],
[ 5.53333275e-01, -5.87763531e-01,  7.62758643e-01,
  3.96171883e-01],
[ 3.10997534e-01, -1.05056946e+00,  1.04708716e+00,
  2.64698913e-01],
[ 2.24968346e+00, -1.24957601e-01,  1.33141568e+00,
  1.44795564e+00],
[ 5.53333275e-01,  8.00654259e-01,  1.04708716e+00,
  1.57942861e+00],
[ 6.74501145e-01,  1.06445364e-01,  9.90221459e-01,
  7.90590793e-01],
[ 1.89829664e-01, -1.24957601e-01,  5.92161531e-01,
  7.90590793e-01],
[ 1.28034050e+00,  1.06445364e-01,  9.33355755e-01,
  1.18500970e+00],
[ 1.03800476e+00,  1.06445364e-01,  1.04708716e+00,
  1.57942861e+00],
[ 1.28034050e+00,  1.06445364e-01,  7.62758643e-01,
  1.44795564e+00],
[-5.25060772e-02, -8.19166497e-01,  7.62758643e-01,
  9.22063763e-01],
```

```
       [ 1.15917263e+00,  3.37848329e-01,  1.21768427e+00,
         1.44795564e+00],
       [ 1.03800476e+00,  5.69251294e-01,  1.10395287e+00,
         1.71090158e+00],
       [ 1.03800476e+00, -1.24957601e-01,  8.19624347e-01,
         1.44795564e+00],
       [ 5.53333275e-01, -1.28197243e+00,  7.05892939e-01,
         9.22063763e-01],
       [ 7.95669016e-01, -1.24957601e-01,  8.19624347e-01,
         1.05353673e+00],
       [ 4.32165405e-01,  8.00654259e-01,  9.33355755e-01,
         1.44795564e+00],
       [ 6.86617933e-02, -1.24957601e-01,  7.62758643e-01,
         7.90590793e-01]])
```

Geraremos a Matriz de Covariânvcia Na diagonal principal aparecem as variâncias das features. Na Standardização, a média já foi para 0 e o desvio padrão para 1. Desvio padrão 1 implica variância 1. Fora da diagonal principal aparecem as covariâncias

[23]:
```
COV=np.cov(XS.T)
COV
```

[23]:
```
array([[ 1.00671141, -0.11010327,  0.87760486,  0.82344326],
       [-0.11010327,  1.00671141, -0.42333835, -0.358937  ],
       [ 0.87760486, -0.42333835,  1.00671141,  0.96921855],
       [ 0.82344326, -0.358937  ,  0.96921855,  1.00671141]])
```

Vamos procurar um novo sistema de eixos que maximizará as variâncias de cada feature E as covariâncias serão anuladas Esse novo sistema de eixos é dado pelos autovetores da matriz de covariância

Utilizaremos Sklearn

[35]:
```
from sklearn.decomposition import PCA #fazendo PCA inicialmente com todas as 4␣
 ↪features
p=PCA(4)
p.fit(XS)
XS4=p.transform(XS)#muda XS para o novo sistema de eixos composto pelos␣
 ↪autovetores
XS4[:7]
```

[35]:
```
array([[-2.26454173,  0.5057039 , -0.12194335, -0.02307332],
       [-2.0864255 , -0.65540473, -0.22725083, -0.10320824],
       [-2.36795045, -0.31847731,  0.05147962, -0.02782523],
       [-2.30419716, -0.57536771,  0.09886044,  0.06631146],
       [-2.38877749,  0.6747674 ,  0.02142785,  0.03739729],
       [-2.07053681,  1.51854856,  0.03068426, -0.00439877],
       [-2.44571134,  0.07456268,  0.34219764,  0.03809657]])
```

[53]:
```python
p
```

[53]:
```
PCA(n_components=4)
```

[26]:
```python
p.explained_variance_#os autovalores
```

[26]:
```
array([2.93035378, 0.92740362, 0.14834223, 0.02074601])
```

[27]:
```python
p.components_
#os autovetores aparecem nas linhas
```

[27]:
```
array([[ 0.52237162, -0.26335492,  0.58125401,  0.56561105],
       [ 0.37231836,  0.92555649,  0.02109478,  0.06541577],
       [-0.72101681,  0.24203288,  0.14089226,  0.6338014 ],
       [-0.26199559,  0.12413481,  0.80115427, -0.52354627]])
```

Questão 1) Mostre que p.explained_variance[0] e p.components_[0] são um par autovalor e autovetor da matriz de covariância Obviamente isso vale para os 4 autovalores e autovetores de COV

[58]:
```python
p.explained_variance_[0], p.components_[0]
```

[58]:
```
(2.9303537755893116,
 array([ 0.52237162, -0.26335492,  0.58125401,  0.56561105]))
```

[67]:
```python
# Demonstração de que p.explained_variance[0] e p.components_[0] são auto valor␣
↪e
# auto vetor da matriz COV
auto_valores, auto_vetores = np.linalg.eig(COV)
auto_valores[0], auto_vetores[0][0], auto_vetores[1][0], auto_vetores[2][0],␣
↪auto_vetores[3][0]
```

[67]:
```
(2.9303537755893183,
 0.5223716204076599,
 -0.26335491531394034,
 0.5812540055976478,
 0.5656110498826492)
```

[55]:
```python
p.explained_variance_ratio_ #observe que com duas das novas features, já temos␣
↪mais de 95% da variância
```

[55]:
```
array([0.72770452, 0.23030523, 0.03683832, 0.00515193])
```

As features são agora ortogonais…nada fora da diagonal principal Vamos observar a nova matriz de covariância COVB (nos novos eixos) as features têm alta variância (diagonal principal) e nenhuma covariância (fora da diagonal) Mais variância, mais informação…assim, com as duas primeiras features (duas primeiras colunas de XS4) já teremos informação suficiente para a visualização, por exemplo. O PCA concentra a variância nas primeiras features

[56]:
```
COVB=np.cov(XS4.T)
COVB
```

[56]:
```
array([[ 2.93035378e+00,  1.16657243e-16,  9.18728181e-16,
        -1.02453467e-17],
       [ 1.16657243e-16,  9.27403622e-01, -1.42480174e-16,
         3.87373066e-17],
       [ 9.18728181e-16, -1.42480174e-16,  1.48342226e-01,
        -4.91776642e-17],
       [-1.02453467e-17,  3.87373066e-17, -4.91776642e-17,
         2.07460140e-02]])
```

Questão 2) Teoricamente, como a Matriz de Covariância é Hermitiana, então os autovetores são ortogonais. Mostre que os 4 autovetores dela são ortogonais entre si (dica..produto interno...).

[77]:
```
p.explained_variance_
ortog = []
ortog.append(np.inner(p.components_[0], p.components_[1]))
ortog.append(np.inner(p.components_[0], p.components_[2]))
ortog.append(np.inner(p.components_[0], p.components_[3]))
ortog.append(np.inner(p.components_[1], p.components_[2]))
ortog.append(np.inner(p.components_[1], p.components_[3]))
ortog.append(np.inner(p.components_[2], p.components_[3]))
```

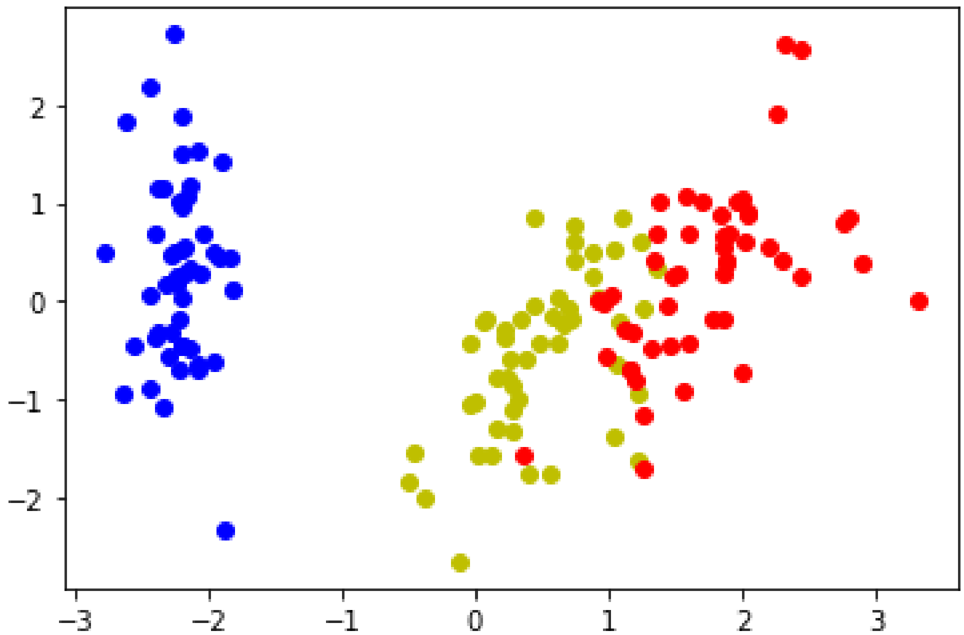
[78]:
```
ortog
```

[78]:
```
[-6.938893903907228e-18,
 5.551115123125783e-17,
 1.6653345369377348e-16,
 2.0816681711721685e-17,
 -4.163336342344337e-17,
 -5.551115123125783e-17]
```

Questão 3) Com as duas primeiras colunas de XS4, faça a visualização do dataset (cada espécie com uma cor)

[79]:
```
esp = df['esp'].copy()
esp = esp.map({'Iris-setosa':0, 'Iris-virginica':2, 'Iris-versicolor':1})
```

[82]:
```
cor = ['bo', 'yo', 'ro']
for i in range(len(XS4)):
    plt.plot(XS4[i,0], XS4[i,1], cor[esp[i]])
plt.show()
```



2
1
0
−1
−2
−3
−2
−1
0
1
2
3