CENTRO UNIVERSITÁRIO FEI

VICTOR BIAZON

RA: 119.115-4

RELATÓRIO III – TÓPICOS ESPECIAIS DE APRENDIZAGEM
LINEAR DISCRIMINANT ANALYSIS

SÃO BERNARDO DO CAMPO

2019

Sumário:

1. Objetivo

Implementar o algoritmo análise de discriminante linear no dataset de flores Iris para comparação com o método do PCA e associação dos dois métodos.

2. Teoria

Análise de discriminante linear

Raschka(2014) descreve a análise de discriminante linear como uma técnica de redução de dimensionalidade utilizada no pré-processamento de dados para serem utilizados em outros algoritmos geralmente para classificação de padrões ou aprendizado de máquina.

O método consiste em calcular a média geral das variáveis independentes e a média das classes, em seguida sendo retirada a média das classes de cada uma delas e sendo calculada a covariância de cada classe. Os parâmetros do LDA são: Scatter Between(Sb) que descreve a distância entre classes, e Scatter Within(Sw) que descreve o espalhamento da própria classe.

O LDA em si calcula um vetor que rotaciona os dados de forma que estes se separem o máximo possível. É possível utiliza em conjunto o PCA com o LDA para primeiramente determinar quais são as variáveis independentes que melhor descrevem a covariância dos dados para ser aplicado o LDA sobre eles e então ser possível a classificação ou avaliação por outros algoritmos.

3. Implementação

A implementação partiu do seguinte fluxo:

Para implementação foram criadas as seguintes funções:

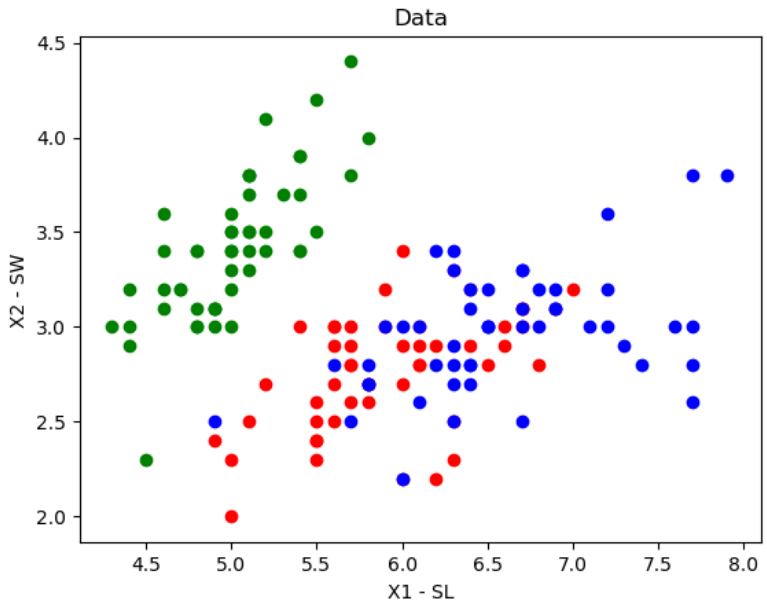
```
def LDA(data):


    #separação das variaveis independentes e dependentes

    X = np.asarray(data.iloc[:,:-1]) #separa as variaveis
independentes no vetor X

    Y = np.asarray(data.iloc[:,-1:]) #separa as variaveis
dependentes no vetor Y


    #Encoding da variavel dependente 1: 'Setosa', 2: 'Versicolor',
3:'Virginica'

    Y, Label_Dict = Encoder(Y)


    #calculo do GrandMean

    GMean =
np.reshape(np.asarray(np.mean(data)),(1,len(np.mean(data))))


    #retirada da media

    M_data = np.copy(X)

    M_data = M_data - GMean



    #separando os dados por classes

    data.set_index("I", inplace=True)

    data.head()
```

```
dataIS = data.loc['Iris-setosa']

dataIVS = data.loc['Iris-versicolor']

dataIVG = data.loc['Iris-virginica']


#calculando a sample Mean

CmeanIS = np.asarray(np.mean(dataIS))

CmeanIVS = np.asarray(np.mean(dataIVS))

CmeanIVG = np.asarray(np.mean(dataIVG))

Cmean = np.vstack((CmeanIS,CmeanIVS,CmeanIVG))


#retirada da media dos dados deparados por classes

M_dataIS = np.asarray(dataIS - CmeanIS)

M_dataIVS = np.asarray(dataIVS - CmeanIVS)

M_dataIVG = np.asarray(dataIVG - CmeanIVG)


#covariancia dos datasets

CovIS = Covariance(M_dataIS)

CovIVS = Covariance(M_dataIVS)

CovIVG = Covariance(M_dataIVG)



#Scatter Between

M_mean = (Cmean - GMean).T

Sb = np.matmul(M_mean,M_mean.T) * len(dataIS)


#Scatter Within

SwIS = (len(dataIS) - 1) * CovIS
```

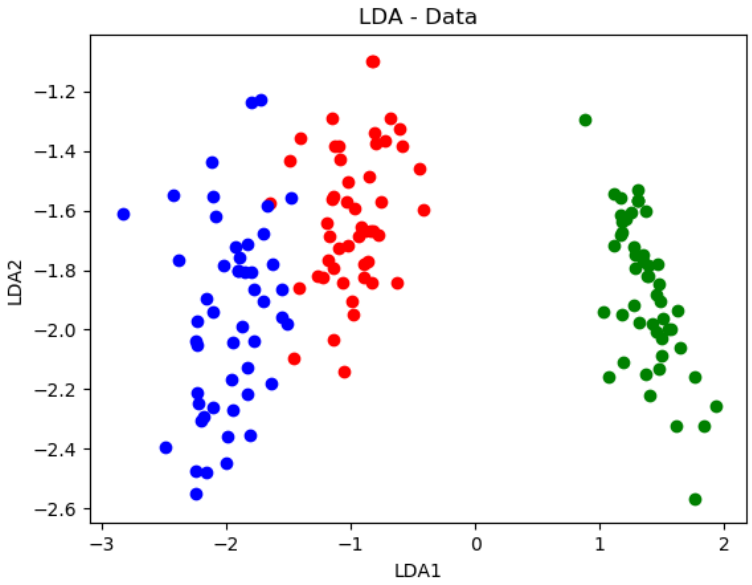
```
    SwIVS = (len(dataIVS) - 1) * CovIVS
    SwIVG = (len(dataIVG) - 1) * CovIVG
    Sw = SwIS + SwIVS + SwIVG

    #Calculando os EigenVectors e EigenValues
    eig_vals, eig_vecs = np.linalg.eig(np.linalg.inv(Sw).dot(Sb))

    #transformando os dados para o LDA em 2 dimensões.
#    Y = X * W
    W = eig_vecs[:,[0,1]]
    Y_n = X.dot(W)
```

4. Resultados

Para testar os algoritmos foi implementado ao dataset de classificação de flores Iris. Com este estudo foi retirado o seguinte resultado.

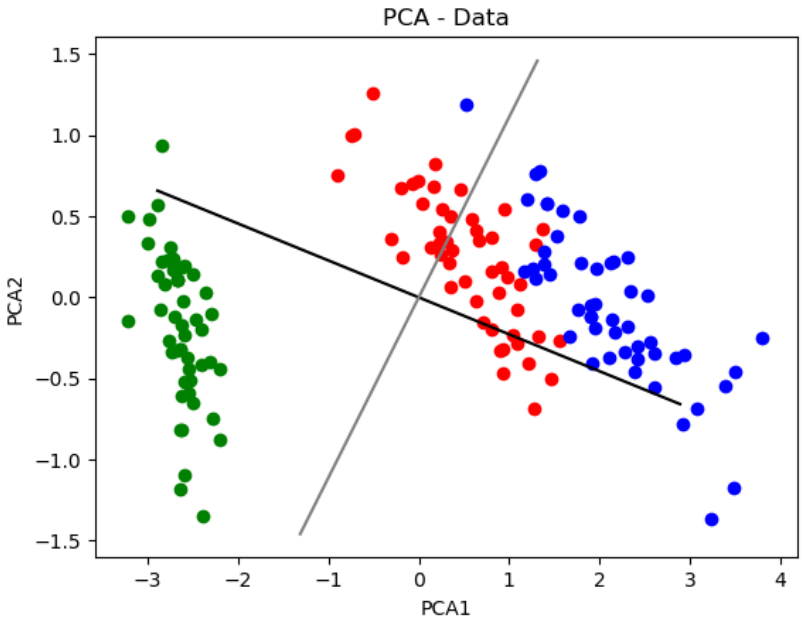


Neste gráfico é mostrado os dados plotados em duas dimensões para representar a separação do dataset inicialmente como disponível nas medições das flores sem qualquer processamento.

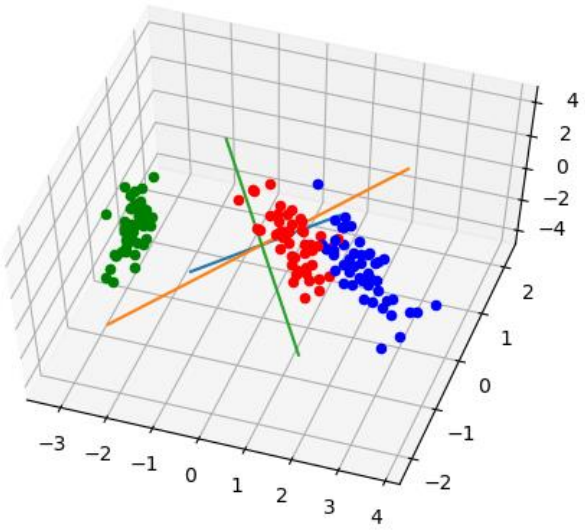


Após o processamento do algoritmo do LDA pode-se verificar claramente a diferenciação das classes em três grupos correspondentes aos três tipos de flores medidos Iris-Setosa, Iris-Versicolor e Iris-Verginica.

Aplicando o PCA ao mesmo dataset é retornado os seguintes gráficos:
Para duas componentes principais:



Para três componentes principais.



Como pode-se notar, não é possível fazer uma boa divisão das classes apenas utilizando as PC's disponíveis pela análise do PCA, portanto é necessário aliá-la ao LDA.

5. Conclusão

Com estes experimentos pode-se verificar que as análises das discriminantes lineares dos datasets descrevem a melhor forma de observar dados para que seja possível realizar a melhor classificação dos mesmos utilizando projeção em relação aos eixos. Ao associar o LDA ao PCA, é possível torna-lo mais eficaz utilizando o PCA para avaliação dos dados e retirada de componentes não relevantes, ou seja, com autovalores estatisticamente não expressivos, e assim reduzindo a dimensão do dataset avaliado pelo LDA.

6. Referências bibliográficas

[1] Sebastian Raschka, **Linear Discriminant Analysis – Bit by bit**, 2014, Disponível em: < https://sebastianraschka.com/Articles/2014_python_lda.html#summarizing-the-lda-approach-in-5-steps >

[2] Lindsay I Smith, **A tutorial on Principal Components Analysis**, 2002 Disponivel em: <ro.umontreal.ca/~pift6080/H09/documents/papers/pca_tutorial.pdf> Acesso: 06/11/2019