P5 - Códigos

Parte A

Carregamento do banco de dados e tratamento dos vetores:

```
colNames = ["X0", "X1", "X2", "X3", "Category"]
permutNames = ["0", "1", "2", "3"]
dataFrame = pd.read_csv("/home/eu/Git/RecPattern/Entrega1/Projeto2/irisDataset/iris.data", header=None, names=colNames)
groups = dataFrame.groupby("Category")
#plotProduct(dataFrame, permutNames, groups)

product = pd.DataFrame(list(itertools.product(permutNames, repeat=2)))
p = product.to_numpy()
p = p.astype(int)

numbers = np.arange(0, len(dataFrame), 1)

dataFrame["Index"] = numbers

aux = dataFrame.to_numpy()

##taking out the label and index
species = pd.DataFrame(data = aux[:, 4], columns = ["species"])
indexes = aux[:, 5]

data = np.delete(aux, 4, 1)
data = np.delete(data, 4, 1)
data = data.astype(np.float64)
```

Cálculo do PCA

```
def pca(data):

    k = np.cov(np.transpose(data))
    eValue, eVector = np.linalg.eig(k)

    normalSort = np.argsort(eValue)

    ###Decreasing sort
    eValue = np.flip(eValue[normalSort])
    eVector = np.flip(eVector[normalSort], axis=0)

    ###New space
    result = []
    for i in range(len(eVector)):
        t = np.transpose(eVector[i])
        result.append(np.array(np.dot(data, t)))

    newData = pd.DataFrame(data = np.transpose(result[:2]), columns=['PC1', 'PC2'])
    newData = pd.concat([newData, species], axis = 1)

    sns.pairplot(newData, hue = 'species', diag_kind=None, palette="YlOrBr")
    plt.show()
    return newData
```

Implementação dos K-vizinhos e matriz de confusão:

```
def knn(data):

    #Separate data
    sh = shuffle(data)
    train = sh[:int(len(sh)/2)].to_numpy()
    test = sh[int(len(sh)/2):].to_numpy()


    #values = data.drop(['species'], axis=1, inplace = False).values

    #Infer category (knn)
    guess = []
```

```
    for i in range(len(test)):
        distLast = 1000000
        for j in range(len(train)):
            dist = distPlain(test[i], train[j])
            if(dist < distLast):
                category = train[j][2]
                distLast = dist
        guess.append([category, test[i][2]])

    #Translate strings into numbers
    spice = ["Iris-versicolor","Iris-virginica", "Iris-setosa"]
    transdic = {"Iris-versicolor":0, "Iris-virginica":1, "Iris-setosa":2}
    for k in range(len(guess)):
        guess[k] = [transdic[spe] for spe in guess[k]]

    #Confusion matrix
    a = np.zeros((3,3))
    for i in range(len(guess)):

        a[guess[i][0], guess[i][1]] += 1

    df_cm = pd.DataFrame(a, index = [i for i in spice],columns = [i for i in spice])
    #plt.figure(figsize = (10,7))
    sns.heatmap(df_cm, annot=True, cmap="gist_gray",  fmt='g', linewidths=.5, cbar=False)
    plt.xlabel("Real")
    plt.ylabel("Inferido")
    plt.tick_params(axis='both', which='major', labelsize=10, labelbottom = False, bottom=False, top = False, labeltop=True)
    plt.title("WithPCA")
    plt.show()


    print(guess)
```