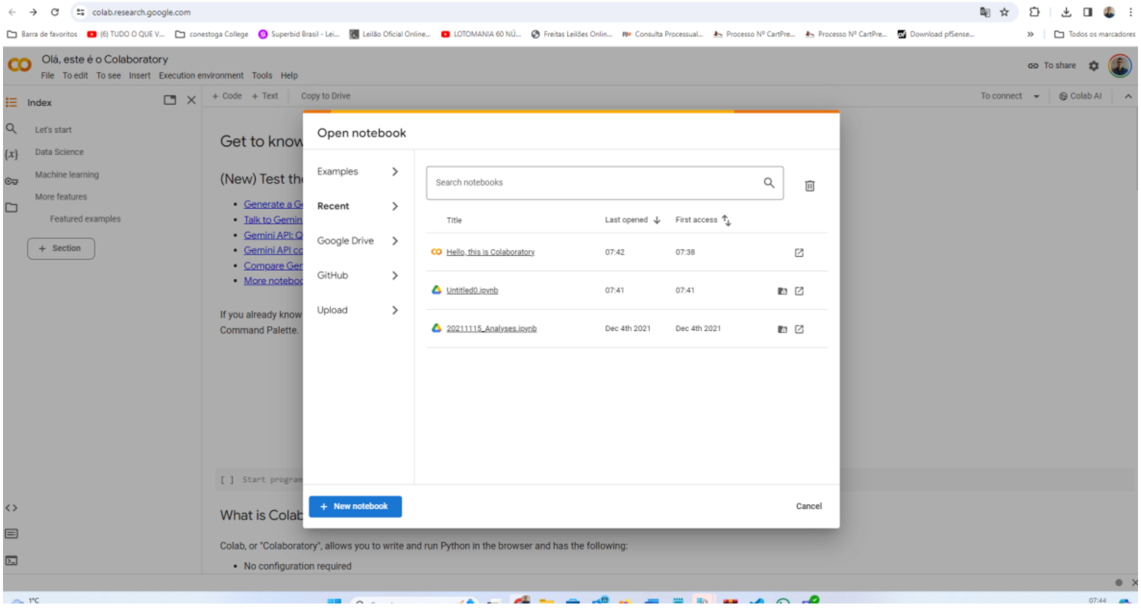
IRIS



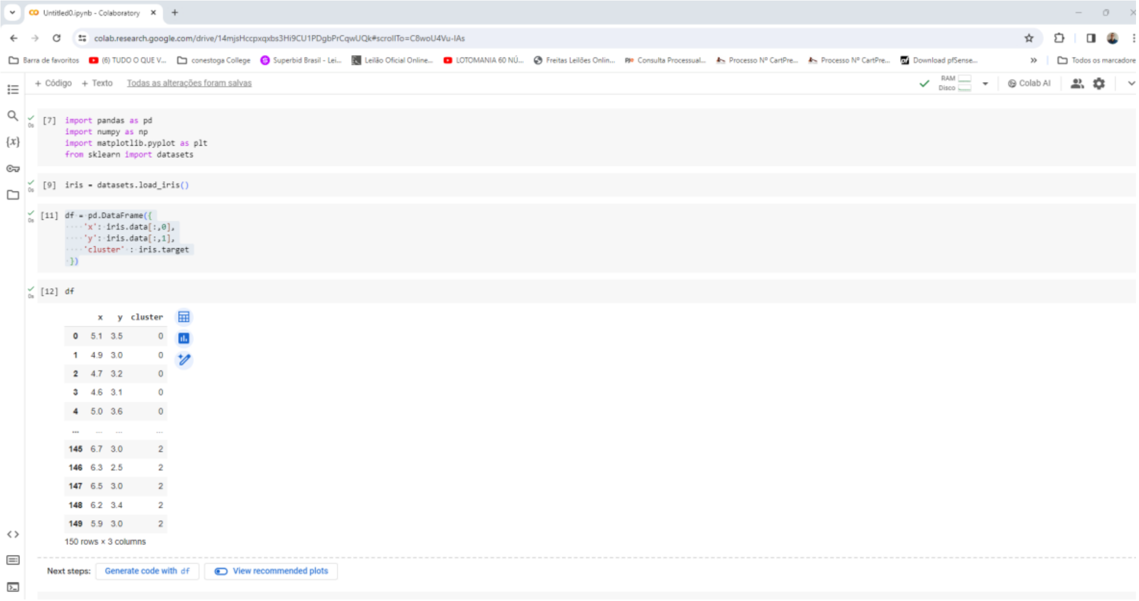
Title	Last opened	First access
Hello, this is Colaboratory	07:42	07:38
Untitled0.ipynb	07:41	07:41
20211115_Analyses.ipynb	Dec 4th 2021	Dec 4th 2021

Colab, or "Colaboratory", allows you to write and run Python in the browser and has the following:

- No configuration required

- Review the details of the IRIS Data set

The IRIS dataset is a classic machine learning and statistics dataset introduced by British biologist and statistician Ronald Fisher in 1936. It contains information about 150 iris flowers from three different species: setosa, versicolor, and virginica. Each iris sample has four features: sepal length, sepal width, petal length, and petal width, all measured in centimetres. The primary purpose of the dataset is to demonstrate various classification techniques. It is commonly used for training machine learning models to classify iris flowers based on their features. The dataset is well-structured, with each species having 50 samples, making it suitable for training and testing algorithms. Due to its simplicity and clarity, it serves as a starting point for many beginners in the field of machine learning. Fisher's work with this dataset pioneered multivariate analysis in biology. It's often utilized in tutorials, textbooks, and introductory courses to illustrate concepts like feature selection, model evaluation, and data visualization. The dataset is freely available and widely used in research and educational contexts across various disciplines, including statistics, machine learning, and botany.



```
[7] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn import datasets
```

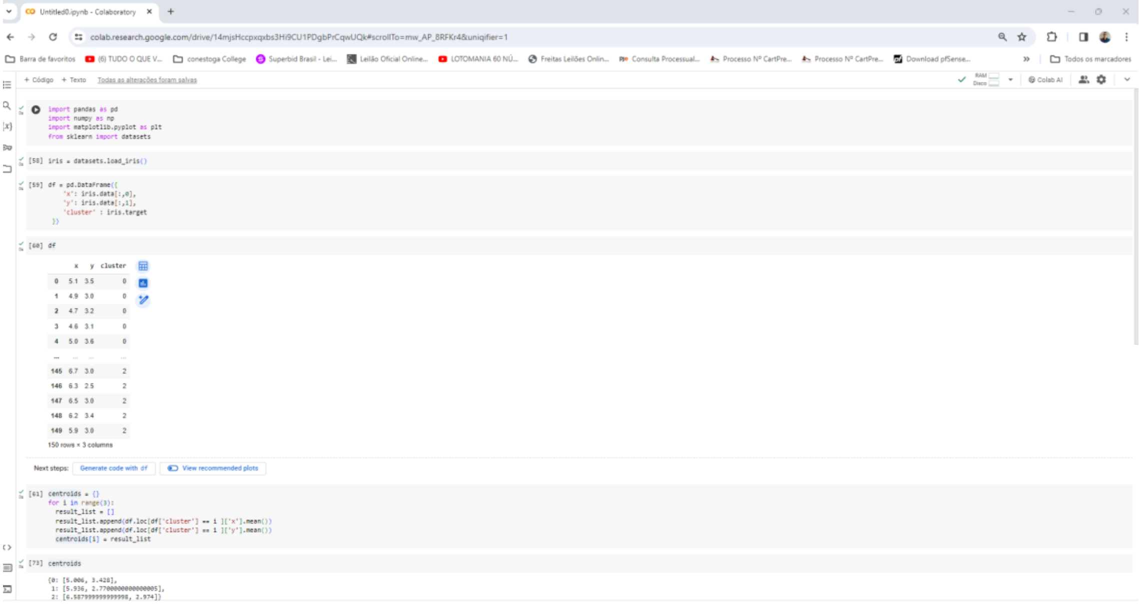
```
[9] iris = datasets.load_iris()
```

```
[11] df = pd.DataFrame({
         'x': iris.data[:,0],
         'y': iris.data[:,1],
         'cluster' : iris.target
      })
```

```
[12] df
```

	x	y	cluster
0	5.1	3.5	0
1	4.9	3.0	0
2	4.7	3.2	0
3	4.6	3.1	0
4	5.0	3.6	0
...	...	...	...
145	6.7	3.0	2
146	6.3	2.5	2
147	6.5	3.0	2
148	6.2	3.4	2
149	5.9	3.0	2

150 rows × 3 columns



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
```

```
[58] iris = datasets.load_iris()
```

```
[59] df = pd.DataFrame({
         'x': iris.data[:,0],
         'y': iris.data[:,1],
         'cluster' : iris.target
      })
```

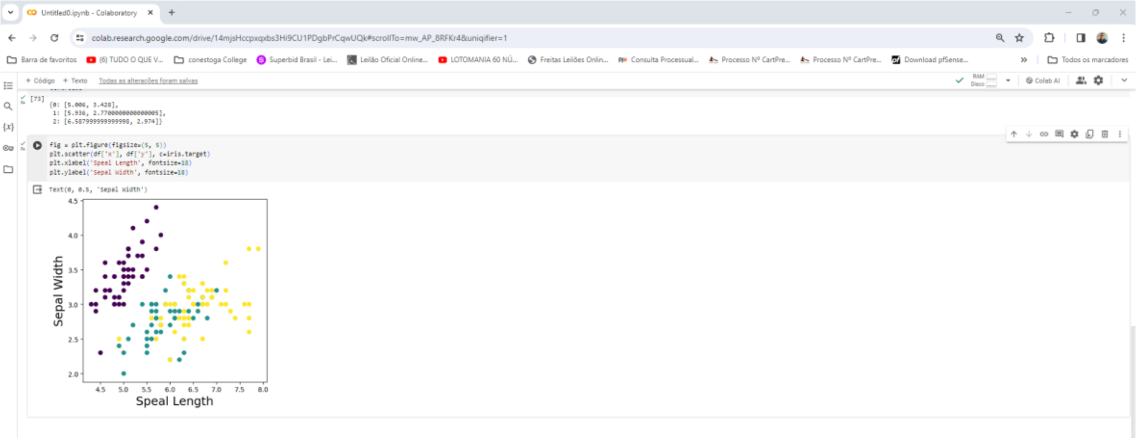
```
[60] df
```

```
       x    y  cluster
0    5.1  3.5        0
1    4.9  3.0        0
2    4.7  3.2        0
3    4.6  3.1        0
4    5.0  3.6        0
...  ...  ...      ...
145  6.7  3.0        2
146  6.3  2.5        2
147  6.5  3.0        2
148  6.2  3.4        2
149  5.9  3.0        2
150 rows × 3 columns
```

```
[61] centroids = {}
     for i in range(3):
       result_list = []
       result_list.append(df.loc[df['cluster'] == i ]['x'].mean())
       result_list.append(df.loc[df['cluster'] == i ]['y'].mean())
       centroids[i] = result_list
```

```
[73] centroids
```

```
{0: [5.006, 3.428],
 1: [5.936, 2.7700000000000005],
 2: [6.587999999999998, 2.974]}
```

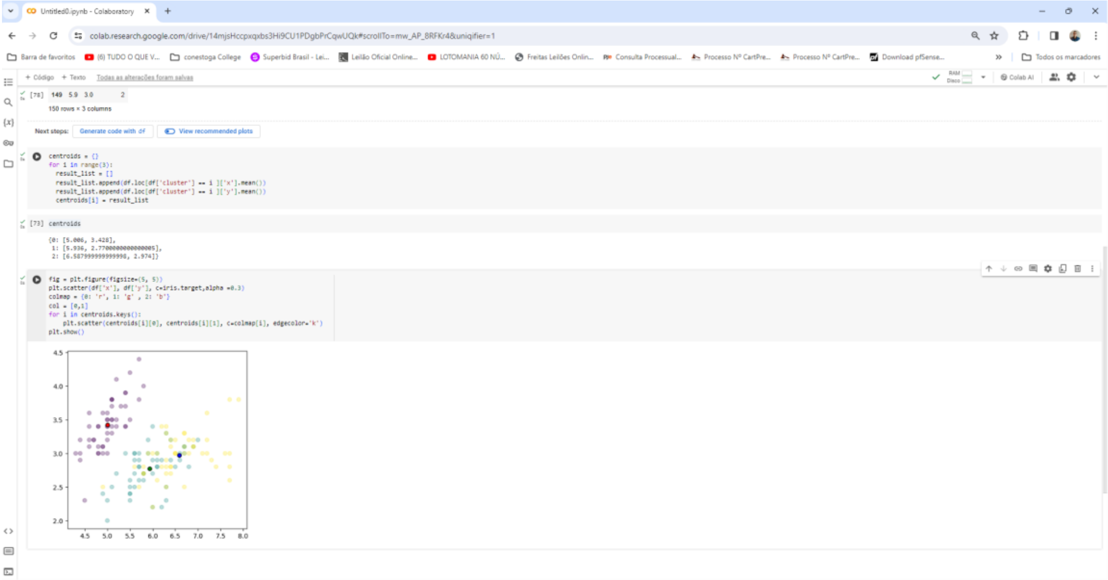

Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=mw_AP_8RFKr4&uniqifier=1
+ Código
+ Texto
Todas as alterações foram salvas

```
[73]
{0: [5.006, 3.428],
 1: [5.936, 2.7700000000000005],
 2: [6.587999999999998, 2.974]}
```

```
fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], c=iris.target)
plt.xlabel('Speal Length', fontsize=18)
plt.ylabel('Sepal Width', fontsize=18)
```

```
Text(0, 0.5, 'Sepal Width')
```

4.5
4.0
3.5
3.0
2.5
2.0
Sepal Width
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Speal Length


Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=mw_AP_8RFKr4&uniqifier=1
+ Código
+ Texto
Todas as alterações foram salvas

```
[78] 149  5.9  3.0        2
150 rows × 3 columns
```

Next steps:
Generate code with df
View recommended plots

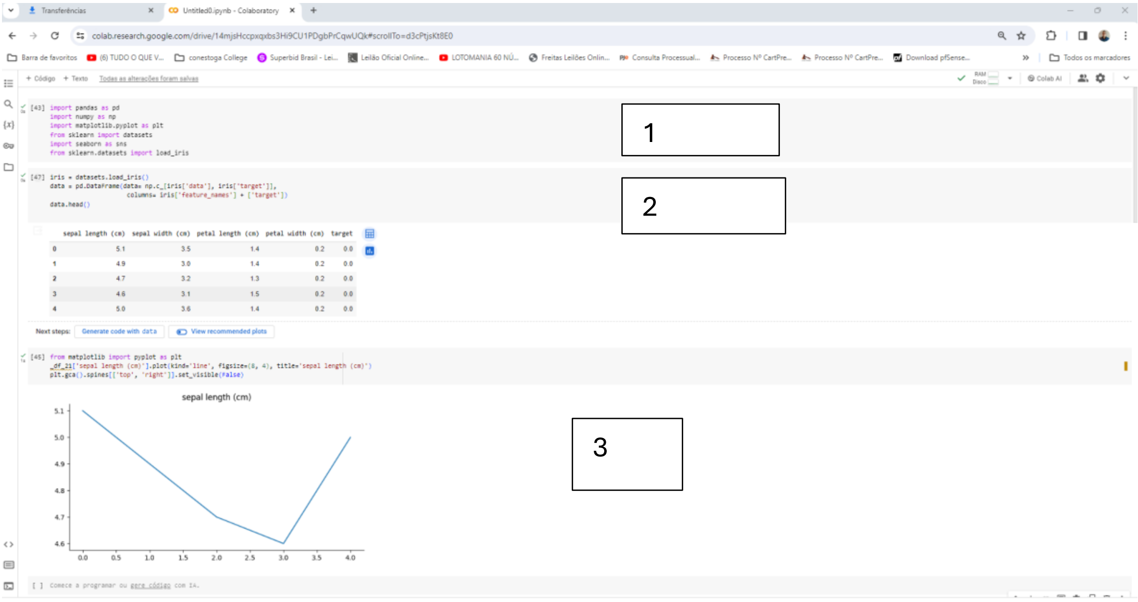
```
centroids = {}
for i in range(3):
  result_list = []
  result_list.append(df.loc[df['cluster'] == i ]['x'].mean())
  result_list.append(df.loc[df['cluster'] == i ]['y'].mean())
  centroids[i] = result_list
```

```
[73] centroids
```

```
{0: [5.006, 3.428],
 1: [5.936, 2.7700000000000005],
 2: [6.587999999999998, 2.974]}
```

```
fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], c=iris.target,alpha =0.3)
colmap = {0: 'r', 1: 'g' , 2: 'b'}
col = [0,1]
for i in centroids.keys():
    plt.scatter(centroids[i][0], centroids[i][1], c=colmap[i], edgecolor='k')
plt.show()
```

Second work.



Transferências
Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=d3cPtjsKt8E0
+ Código
+ Texto
Todas as alterações foram salvas
1
2
3

```
[43] import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn import datasets
     import seaborn as sns
     from sklearn.datasets import load_iris
```

```
[47] iris = datasets.load_iris()
     data = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
                         columns= iris['feature_names'] + ['target'])
     data.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

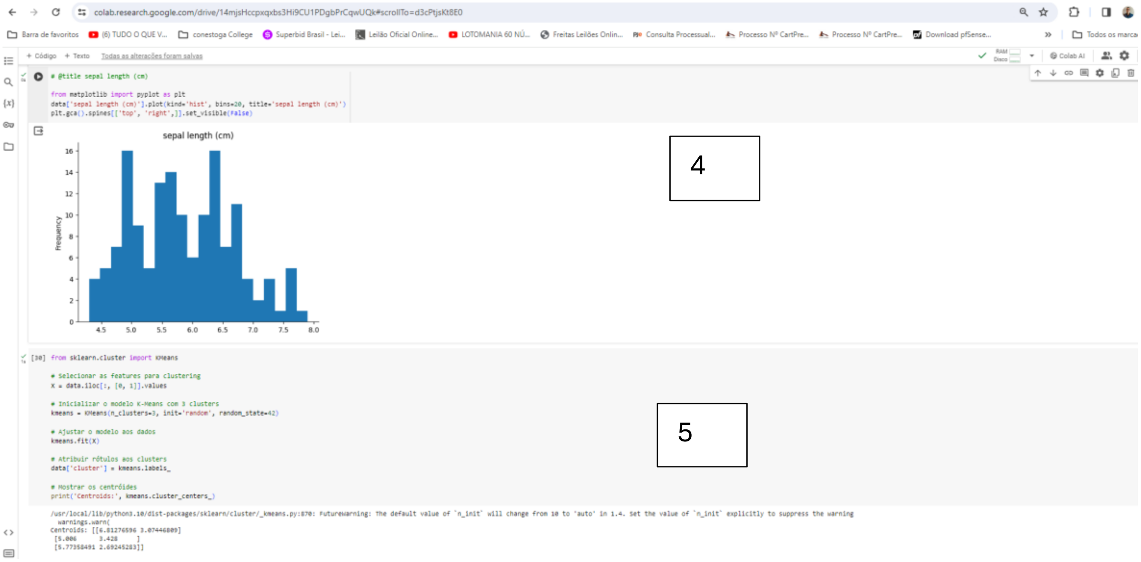
Next steps: Generate code with data | View recommended plots

```
[45] from matplotlib import pyplot as plt
     _df_21['sepal length (cm)'].plot(kind='line', figsize=(8, 4), title='sepal length (cm)')
     plt.gca().spines[['top', 'right']].set_visible(False)
```

1- Importing Libraries

2- Creating Data frame structure

3- Exploratory analysis of sepal length and sepal width for the 3 classes



colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=d3cPtjsKt8E0
+ Código
+ Texto
Todas as alterações foram salvas
4
5

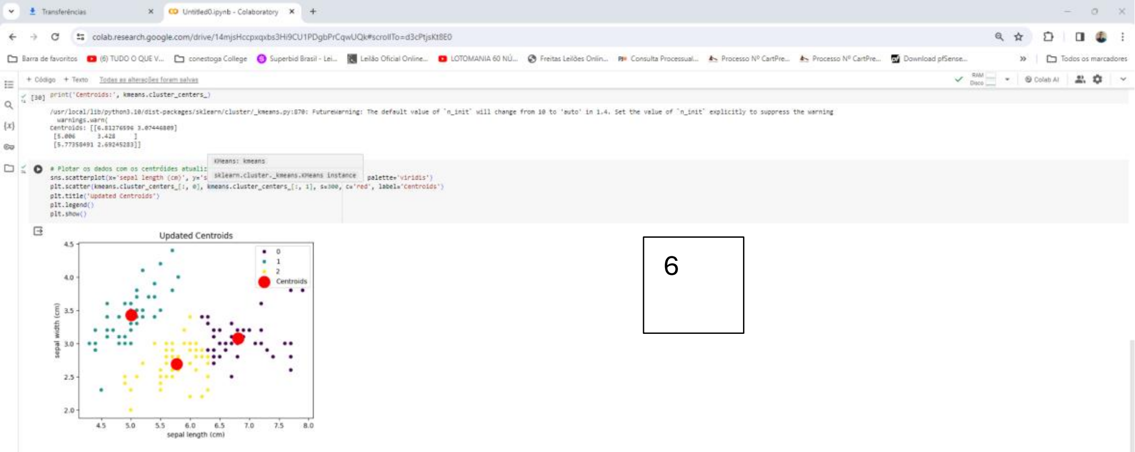
```
# @title sepal length (cm)

from matplotlib import pyplot as plt
data['sepal length (cm)'].plot(kind='hist', bins=20, title='sepal length (cm)')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

sepal length (cm)
Frequency
0
2
4
6
8
10
12
14
16
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

```
[30] from sklearn.cluster import KMeans

     # Selecionar as features para clustering
     X = data.iloc[:, [0, 1]].values

     # Inicializar o modelo K-Means com 3 clusters
     kmeans = KMeans(n_clusters=3, init='random', random_state=42)

     # Ajustar o modelo aos dados
     kmeans.fit(X)

     # Atribuir rótulos aos clusters
     data['cluster'] = kmeans.labels_

     # Mostrar os centróides
     print('Centroids:', kmeans.cluster_centers_)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
Centroids: [[6.81276596 3.07446809]
 [5.006      3.428     ]
 [5.77358491 2.69245283]]
```

5 Creating centroids

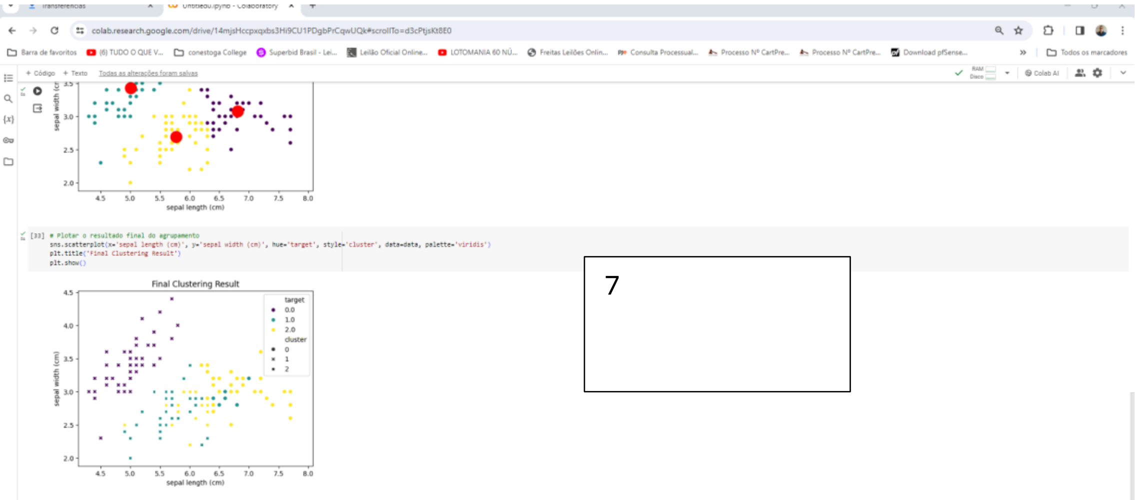

Transferências
Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=d3cPtjsKt8E0
+ Código + Texto Todas as alterações foram salvas
KMeans: kmeans
sklearn.cluster._kmeans.KMeans instance
Updated Centroids
sepal width (cm)
sepal length (cm)
4.5
4.0
3.5
3.0
2.5
2.0
4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
0
1
2
Centroids

```
[30] print('Centroids:', kmeans.cluster_centers_)

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
Centroids: [[6.81276596 3.07446809]
 [5.006      3.428     ]
 [5.77358491 2.69245283]]
```

```
# Plotar os dados com os centróides atualiz
sns.scatterplot(x='sepal length (cm)', y='s                                   palette='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', label='Centroids')
plt.title('Updated Centroids')
plt.legend()
plt.show()
```

6

6- update cetroids


Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/14mjsHccpxqxbs3Hi9CU1PDgbPrCqwUQk#scrollTo=d3cPtjsKt8E0
+ Código + Texto Todas as alterações foram salvas
Final Clustering Result
sepal width (cm)
sepal length (cm)
4.5
4.0
3.5
3.0
2.5
2.0
4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
target
0.0
1.0
2.0
cluster
0
1
2

```
[33] # Plotar o resultado final do agrupamento
sns.scatterplot(x='sepal length (cm)', y='sepal width (cm)', hue='target', style='cluster', data=data, palette='viridis')
plt.title('Final Clustering Result')
plt.show()
```

7

7- Final clustering result

In this assignment, I conducted an exploratory analysis of a data set that included measurements of the length and width of sepals from three different classes of flowers. After importing the libraries and enabling and structuring the data into a Data Frame, I explored the distribution of sepal measurements for each flower class.

Then, I applied the K-means clustering algorithm to group the data into clusters based on their characteristics. I created initial centroids and assigned the data to the closest clusters. Iteratively, I was able to update the centroids and recalculate the assignment of the data to the clusters until the centroids stabilized.

We visualize the updated centroids with the original data to better understand the distribution of the clusters. Finally, I presented the final clustering result graphically, highlighting the clusters identified by the K-means algorithm. However, I would like to see a graphic image in another way, but I could not.

Reference

https://colab.research.google.com

https://www.kaggle.com/datasets/uciml/irishttps://medium.com/@sarakarim/k-means-clustering-for-iris-dataset-in-google-colab-30a6d78556c4

https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html