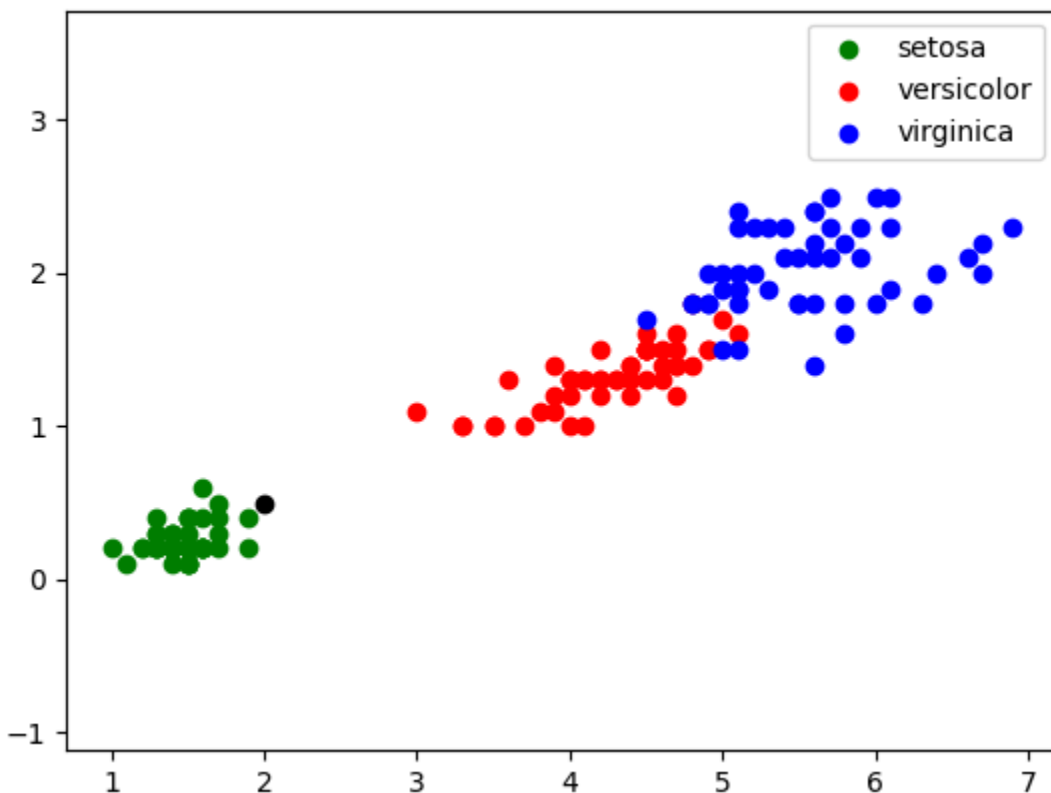


Algorithmes KNN (k plus proches des voisins)

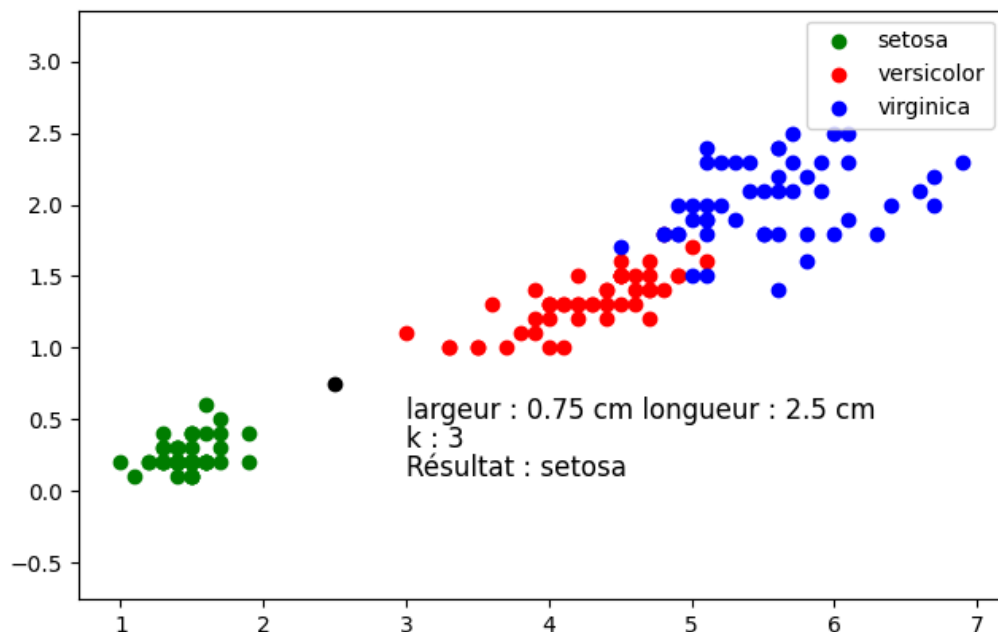
À faire vous-même 1

```
1
2 import pandas
3 import matplotlib.pyplot as plt
4
5 iris=pandas.read_csv("iris.csv")
6 x=iris.loc[:, "petal_length"]
7 y=iris.loc[:, "petal_width"]
8 lab=iris.loc[:, "species"]
9
10 plt.axis('equal')
11 plt.scatter(x[lab == 0], y[lab == 0], color='g', label='setosa')
12 plt.scatter(x[lab == 1], y[lab == 1], color='r', label='versicolor')
13 plt.scatter(x[lab == 2], y[lab == 2], color='b', label='virginica')
14 plt.scatter(2.0, 0.5, color='k')
15 plt.legend()
16 plt.show()
```

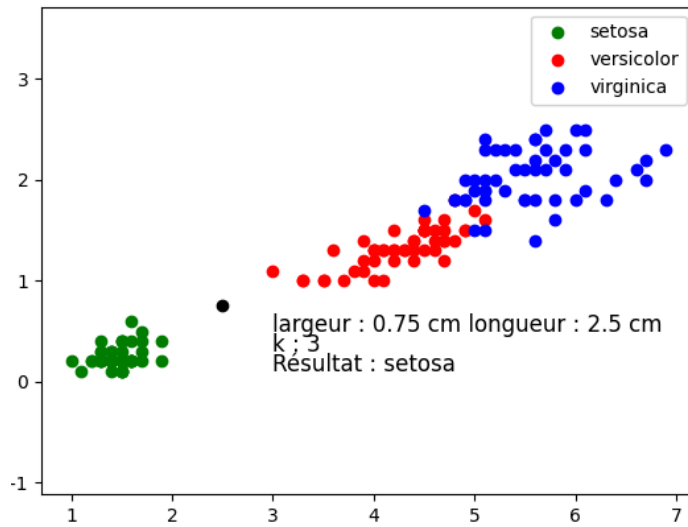


À faire vous-même 2

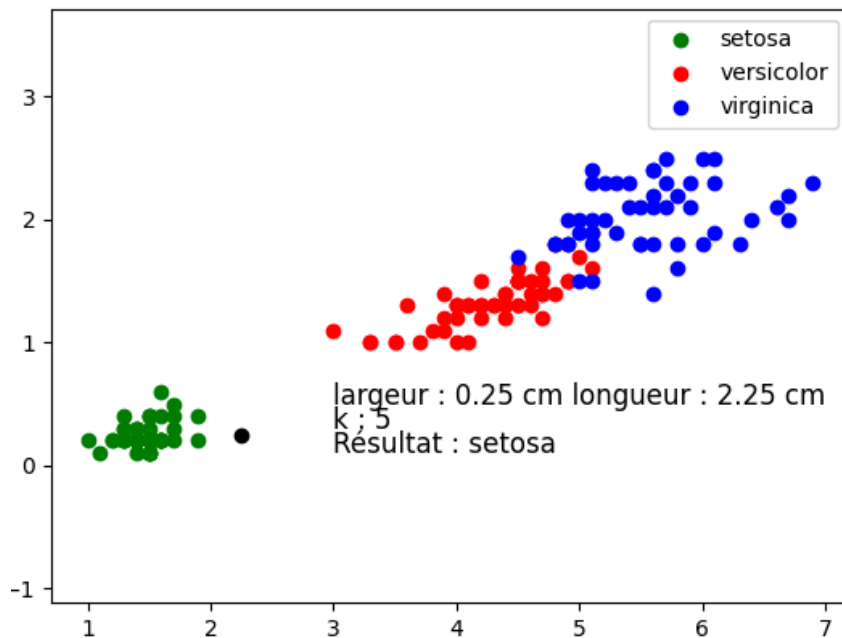
```
1
2 import pandas
3 import matplotlib.pyplot as plt
4 from sklearn.neighbors import KNeighborsClassifier
5
6 #traitement CSV
7 iris=pandas.read_csv("iris.csv")
8 x=iris.loc[:, "petal_length"]
9 y=iris.loc[:, "petal_width"]
10 lab=iris.loc[:, "species"]
11 #fin traitement CSV
12
13 #valeurs
14 longueur=2.5
15 largeur=0.75
16 k=3
17 #fin valeurs
18
19 #graphique
20 plt.axis('equal')
21 plt.scatter(x[lab == 0], y[lab == 0], color='g', label='setosa')
22 plt.scatter(x[lab == 1], y[lab == 1], color='r', label='versicolor')
23 plt.scatter(x[lab == 2], y[lab == 2], color='b', label='virginica')
24 plt.scatter(longueur, largeur, color='k')
25 plt.legend()
26 #fin graphique
27
28 #algo knn
29 d=list(zip(x,y))
30 model = KNeighborsClassifier(n_neighbors=k)
31 model.fit(d,lab)
32 prediction= model.predict([[longueur,largeur]])
33 #fin algo knn
34
35 #affichage résultats
36 txt="Résultat : "
37 if prediction[0]==0:
38     txt=txt+"setosa"
39 if prediction[0]==1:
40     txt=txt+"versicolor"
41 if prediction[0]==2:
42     txt=txt+"virginica"
43 plt.text(3.0, 0.5, f"largeur : {largeur} cm longueur : {longueur} cm", fontsize=12)
44 plt.text(3.0, 0.3, f"k : {k}", fontsize=12)
45 plt.text(3.0, 0.1, txt, fontsize=12)
46 #fin affichage résultats
47
```



À faire vous-même 3



À faire vous-même 4



La largeur: 0.25cm et longueur: 2cm