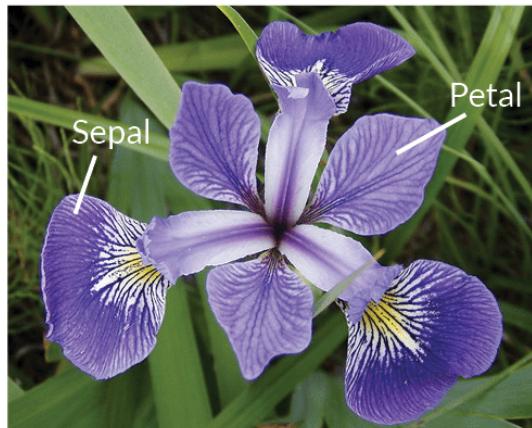


```
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
import seaborn as sns
from sklearn.datasets import load_iris
```

```
iris = load_iris()
```



**Iris Versicolor**




**Iris Setosa**



**Iris Virginica**

```
data = pd.DataFrame(data=iris.data, columns=iris.feature_names)
data.head()
```

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

```
data['target'] = iris.target
```

```
data
```

```
data[data.target == 1].head()
```

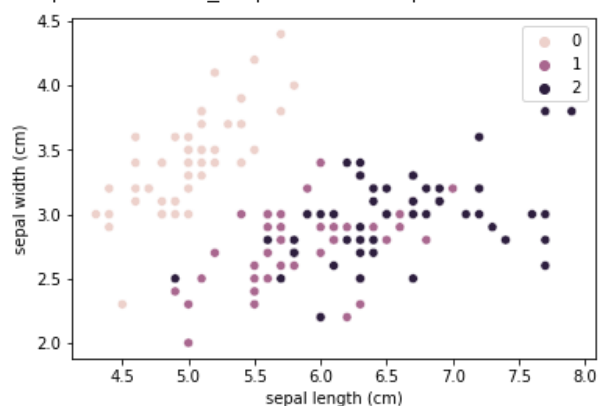
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
50	7.0	3.2	4.7	1.4	1
51	6.4	3.2	4.5	1.5	1
52	6.9	3.1	4.9	1.5	1
53	5.5	2.3	4.0	1.3	1
54	6.5	2.8	4.6	1.5	1
146	6.3	2.5	5.0	1.9	2

```
data[data.target == 1].shape
```

```
(50, 5)
```

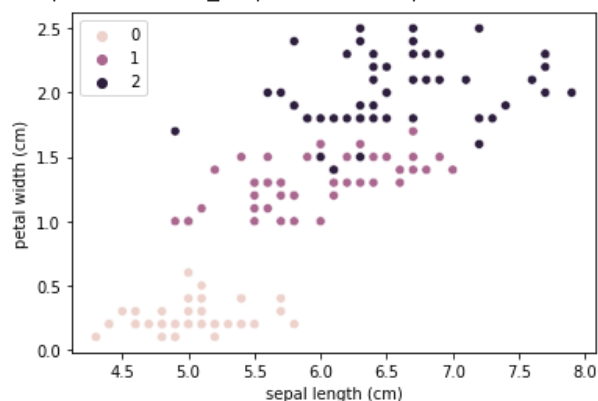
```
sns.scatterplot(data=data, x=data['petal length (cm)'], y=data['sepal width (cm)'], hue=data.target.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcd06e72910>
```



```
sns.scatterplot(data=data, x=data['sepal length (cm)'], y=data['petal width (cm)'], hue=data.target.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcd06813a90>
```



```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(data[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
data.target], test_size=0.2, random_state=3)
```

```
X_train
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	
14	5.8	4.0	1.2	0.2	
24	4.8	3.4	1.9	0.2	
139	6.9	3.1	5.4	2.1	
58	6.6	2.9	4.6	1.3	
8	4.4	2.9	1.4	0.2	
...	...	...	...	...	
107	7.3	2.9	6.3	1.8	
21	5.1	3.7	1.5	0.4	
0	5.1	3.5	1.4	0.2	

y\_train

```

14    0
24    0
139   2
58    1
8     0
      ..
107   2
21    0
0     0
131   2
106   2
Name: target, Length: 120, dtype: int64

```

```
from sklearn.metrics import accuracy_score
```

```

for k in range(1, 20, 2):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    print(k, " :", accuracy_score(y_test, knn.predict(X_test)))

```

```

1  : 0.9333333333333333
3  : 0.9666666666666667
5  : 0.9666666666666667
7  : 0.9666666666666667
9  : 1.0
11 : 1.0
13 : 1.0
15 : 1.0
17 : 1.0
19 : 1.0

```

```
from sklearn.metrics import confusion_matrix
```

```

knn = KNeighborsClassifier(9)
knn.fit(X_train, y_train)
cm = confusion_matrix(y_test, knn.predict(X_test))

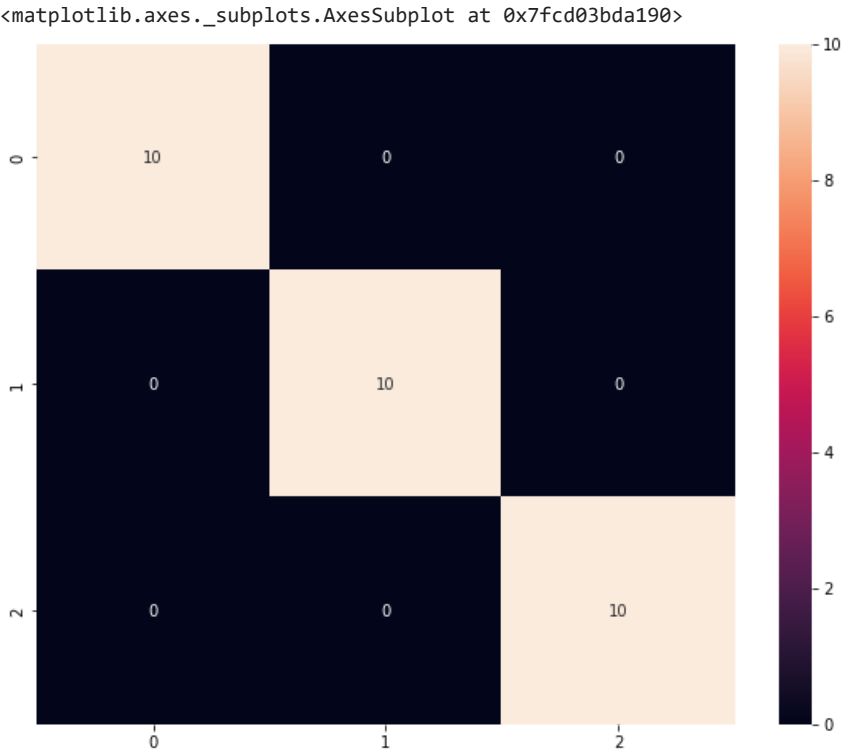
```

```
import matplotlib.pyplot as plt
```

```

plt.figure(figsize=(10,8))
sns.heatmap(cm, annot=True)

```



[Платные продукты Colab](#) - [Отменить подписку](#)

✓ 0 сек. выполнено в 18:48

