

در مرحله اول درست بودن dataset چک شد

```
[3] 1 data_iris = sb.load_dataset("iris")
    2
    3 print("5 first elements")
    4 data_iris.head()
```

5 first elements

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
[4] 1 print("5 last elements")
    2 data_iris.tail()
```

5 last elements

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

در مرحله ی بعد اطلاعات آماری dataset مشاهده می شود

Iris Info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   sepal_length    150 non-null    float64
1   sepal_width     150 non-null    float64
2   petal_length    150 non-null    float64
3   petal_width     150 non-null    float64
4   species         150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Description

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Some statistical info:

<< Setosa >>

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.00000	50.00000	50.00000
mean	5.00600	3.42800	1.46200	0.24600
std	0.35249	0.37906	0.17366	0.10538
min	4.30000	2.30000	1.00000	0.10000
25%	4.80000	3.20000	1.40000	0.20000
50%	5.00000	3.40000	1.50000	0.20000
75%	5.20000	3.67500	1.57500	0.30000
max	5.80000	4.40000	1.90000	0.60000

<< Versicolor >>

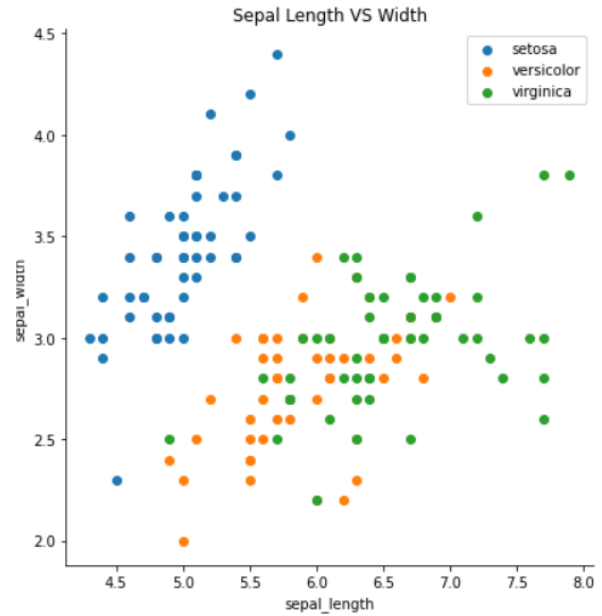
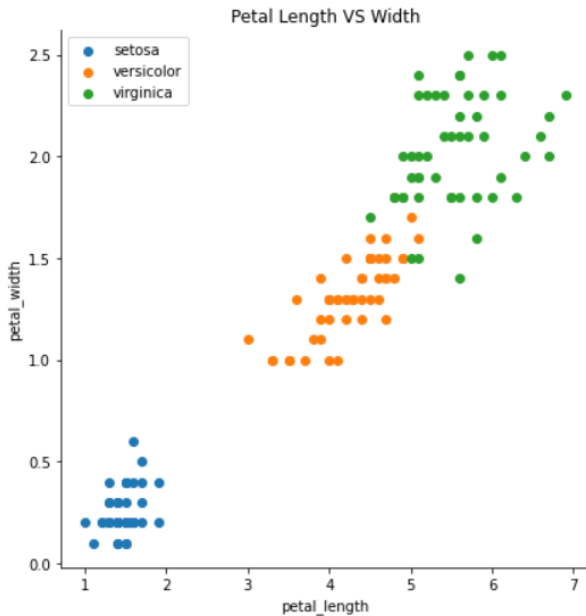
	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.00000	50.00000	50.00000
mean	5.93600	2.77000	4.26000	1.32600
std	0.51617	0.31379	0.46991	0.19775
min	4.90000	2.00000	3.00000	1.00000
25%	5.60000	2.52500	4.00000	1.20000
50%	5.90000	2.80000	4.35000	1.30000
75%	6.30000	3.00000	4.60000	1.50000
max	7.00000	3.40000	5.10000	1.80000

<< Virginica >>

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.00000	50.00000	50.00000
mean	6.58800	2.97400	5.55200	2.02600
std	0.63588	0.32249	0.55189	0.27465
min	4.90000	2.20000	4.50000	1.40000
25%	6.22500	2.80000	5.10000	1.80000
50%	6.50000	3.00000	5.55000	2.00000
75%	6.90000	3.17500	5.87500	2.30000
max	7.90000	3.80000	6.90000	2.50000

نمودار تفکیک بر اساس Sepal و Petal:

petal and sepal length
Text(0.5, 1.0, 'Sepal Length VS Width')



تفکیک بر اساس Petal بهتر است چون همان طور که در شکل می بینیم رنگ های مختلف در یک دسته قرار دارند و دقت regression بر اساس این ویژگی بهتر و دسته بندی بر اساس Petal دقیق تر است

Test Accuracy (Scikit-Learn): 94.33333333333333 %

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	9
versicolor	0.83	1.00	0.91	10
virginica	1.00	0.82	0.90	11
accuracy			0.93	30
macro avg	0.94	0.94	0.94	30
weighted avg	0.94	0.93	0.93	30

نکته: تابع logistic regression کتابخانه ی scikit-learn پایتون بر اساس مدل One vs rest مدل سازی و دسته بندی می کند و برای میانگین گیری از دو روش macro و weighted استفاده شده است

در شکل زیر تمام حالت های ممکن دسته بندی با کمک کتابخانه ی seaborn رسم شده است:

