

# Iris-solution

2019년 3월 21일 목요일 오전 11:47

## 1. Defining the problem statement :

- 붓꽃의 종 파악하기

## 2. Collecting the data :

## 3. Exploratory data analysis :

```
In [69]: 1 a = iris_dataset['data']  
         2 # print(a)
```

```
In [68]: 1 b = iris_dataset['target']  
         2 # print(b)
```

```
In [65]: 1 da= pd.DataFrame(a)  
         2 # print(da)
```

```
In [61]: 1 db= pd.DataFrame(b)
```

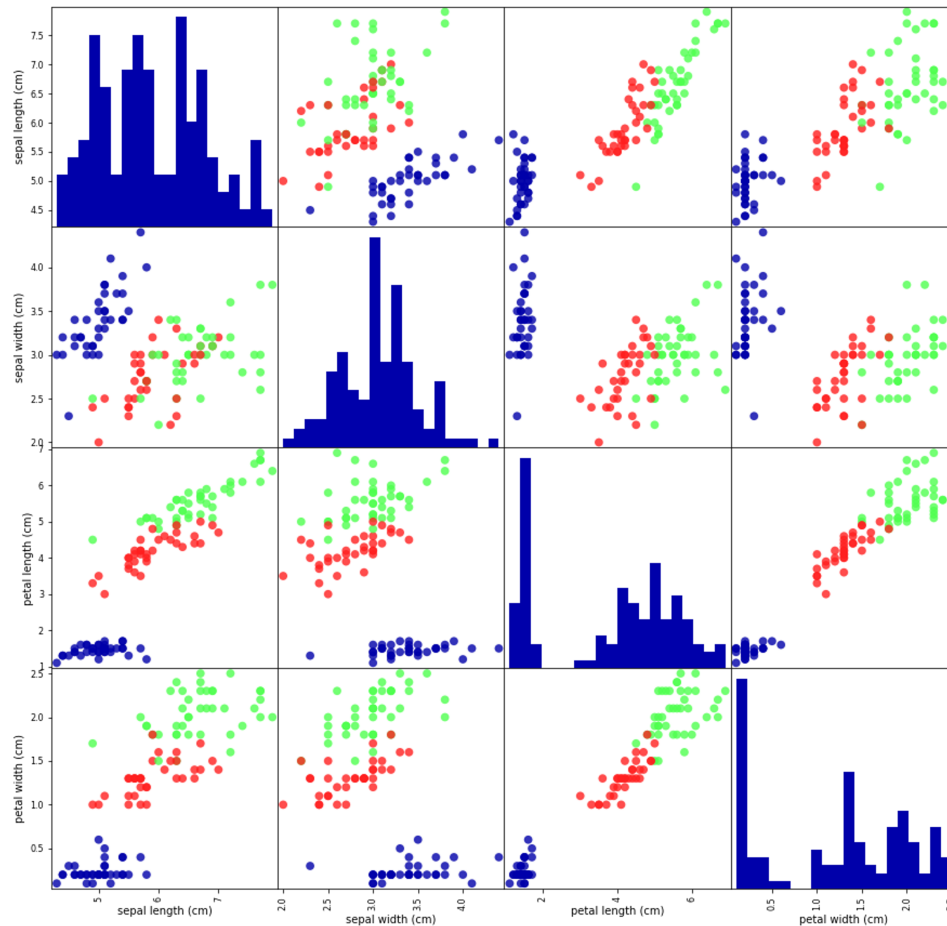
```
In [62]: 1 df = pd.concat([da, db], axis=1)
```

```
In [64]: 1 df.head()
```

```
Out[64]:
```

	0	1	2	3	0
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

## 4. Feature engineering :



## 5. Modelling :

첫 번째 머신 러닝 모델: k-최근접 이웃 알고리즘

```
In [25]: 1 from sklearn.neighbors import KNeighborsClassifier
         2 knn = KNeighborsClassifier(n_neighbors=1)
```

```
In [26]: 1 knn.fit(X_train, y_train)
```

```
Out[26]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                             metric_params=None, n_jobs=None, n_neighbors=1, p=2,
                             weights='uniform')
```

## 6. Testing :

첫 번째 머신 러닝 모델: k-최근접 이웃 알고리즘

```
In [25]: 1 from sklearn.neighbors import KNeighborsClassifier
         2 knn = KNeighborsClassifier(n_neighbors=1)
```

```
In [26]: 1 knn.fit(X_train, y_train)
```

```
Out[26]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                               metric_params=None, n_jobs=None, n_neighbors=1, p=2,
                               weights='uniform')
```

모델 평가하기

```
In [29]: 1 y_pred = knn.predict(X_test)
         2 print("테스트 세트에 대한 예측값:\n", y_pred)
```

테스트 세트에 대한 예측값:

```
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2]
```

```
In [30]: 1 print("테스트 세트의 정확도: {:.2f}".format(np.mean(y_pred == y_test)))
```

테스트 세트의 정확도: 0.97

```
In [31]: 1 print("테스트 세트의 정확도: {:.2f}".format(knn.score(X_test, y_test)))
```

테스트 세트의 정확도: 0.97

요약

```
In [32]: 1 X_train, X_test, y_train, y_test = train_test_split(
         2     iris_dataset['data'], iris_dataset['target'], random_state=0)
         3
         4 knn = KNeighborsClassifier(n_neighbors=1)
         5 knn.fit(X_train, y_train)
         6
         7 print("테스트 세트의 정확도: {:.2f}".format(knn.score(X_test, y_test)))
```

테스트 세트의 정확도: 0.97

