```
import sys
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import scipy as sp
import seaborn as sns
```

```
from sklearn.datasets import load_iris
iris = load_iris()
```

```
iris.keys()
```

```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

```
iris.target.shape
iris.data.shape
```

```
(150, 4)
```

```
iris['feature_names']  #칼럼명 확인
```

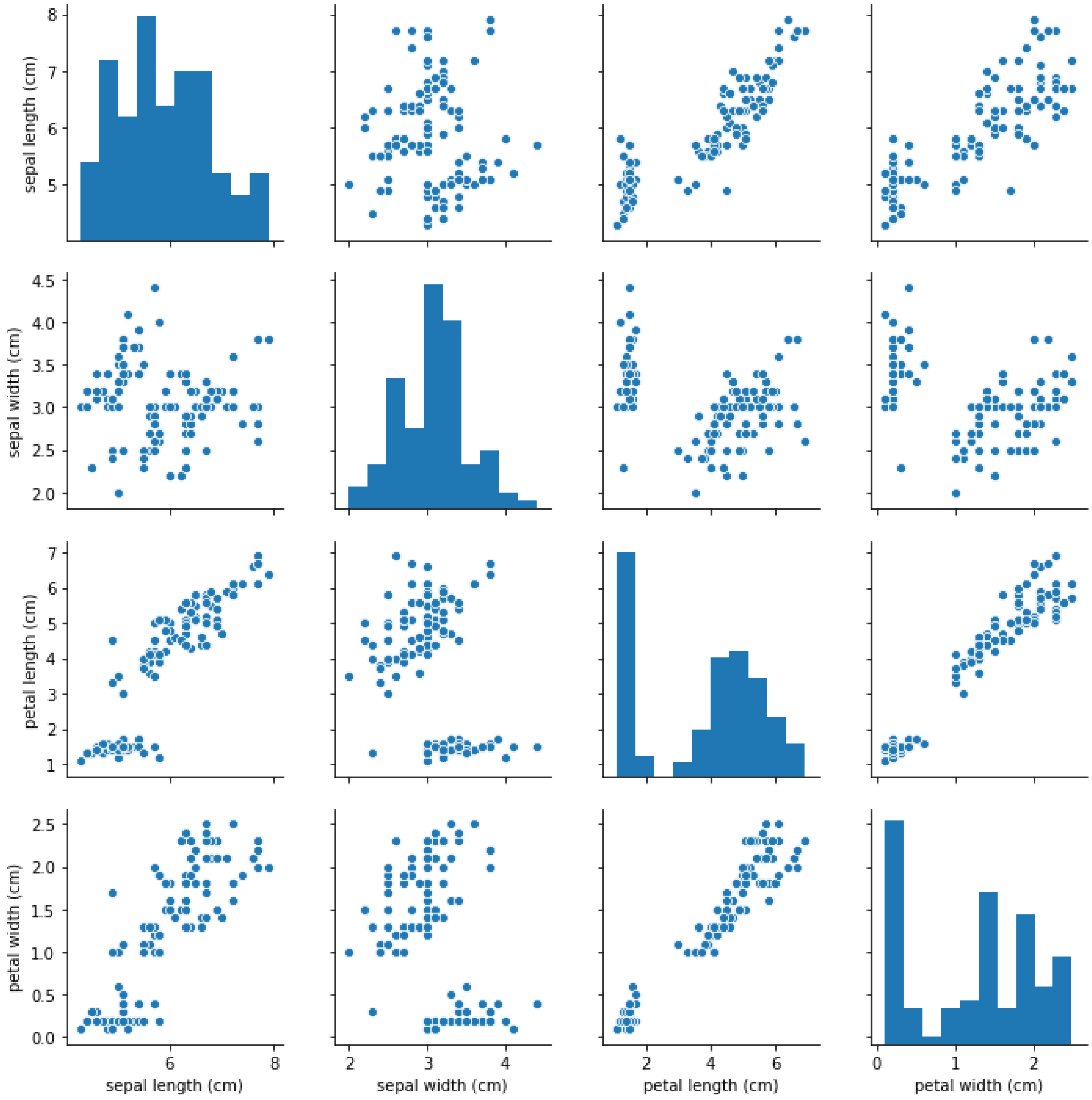
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

```
# 훈련 데이터와 테스트 데이터 나누기
# train_test_split (문제, 답, random_state = 0)  random_state는 패턴을 고정시켜주는 값
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(iris['data'],
                                                    iris['target'],
                                                    random_state = 0)
```

```
## 시각화
# DataFrame에 들어가야 하는 데이터 타입 dictionary
# Series에 들어가는 데이터 타입 list

iris_df = pd.DataFrame(X_train, columns=iris.feature_names)
iris_df['y'] = Y_train
iris_df['y'] = iris_df['y'].astype('category')   #범주형으로 타입 변경
```

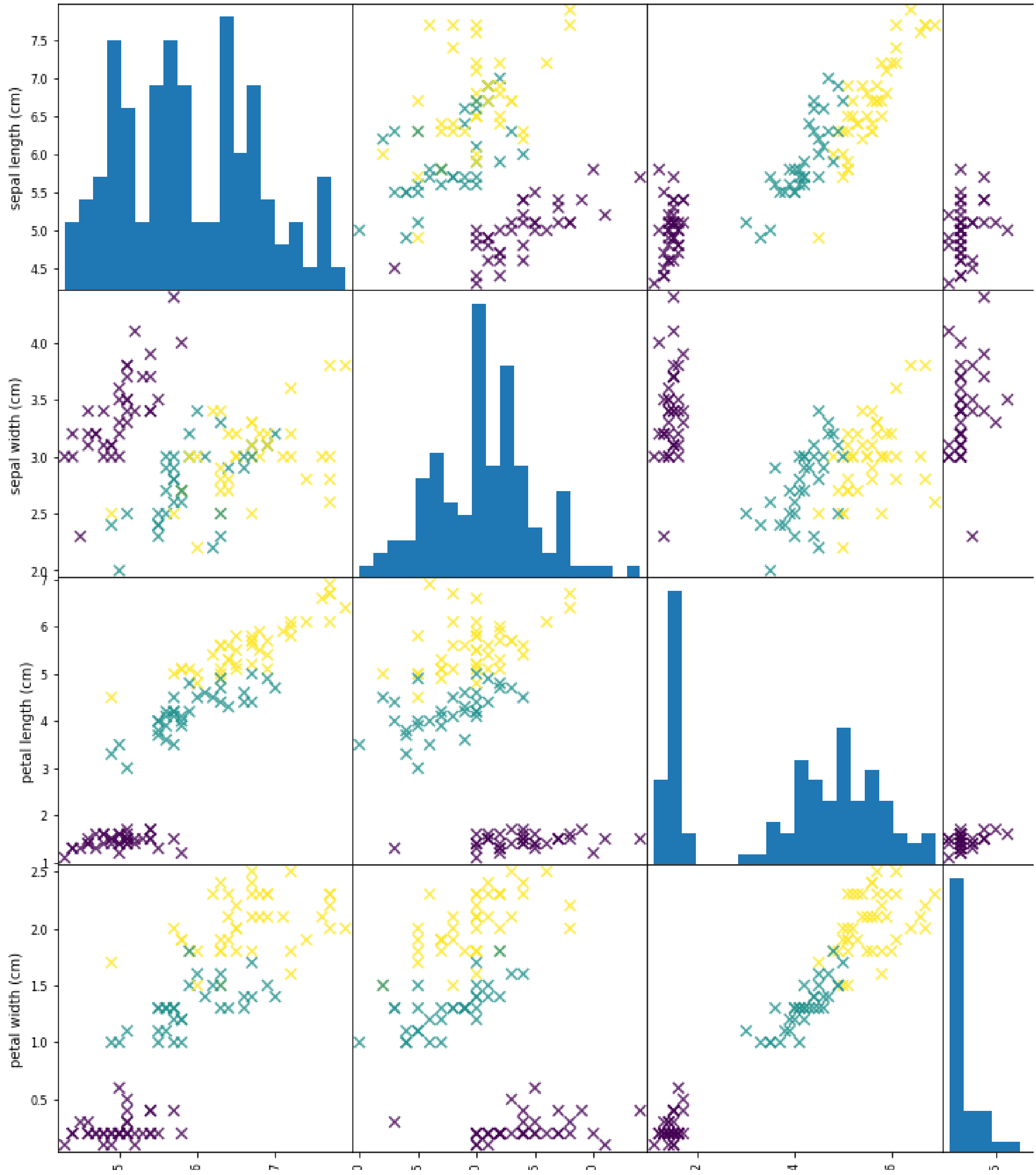
```
sns.pairplot(iris_df.iloc[:,0:4])

# iloc : 인덱스로 선택
# loc : 칼럼으로 선택
```

```
<seaborn.axisgrid.PairGrid at 0x2f64762ecf8>
```



```
# 판다스 플로팅 기능
pd.plotting.scatter_matrix(iris_df, c=Y_train, #색
                           figsize =(15,15),
                           marker='x',
                           hist_kwds={'bins':20}, #막대의 개수
                           s=60,
                           alpha=0.8) #투명도
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000002F647FBCFD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F648065F60>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F64852FB00>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F64856D0F0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002F64859C6A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F6485CEC50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F64860D240>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F64863D828>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002F64863D860>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F6486AC390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F6486DE940>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F648712EF0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000002F64874D4E0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F648780A90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F6487C0080>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000002F6487EF630>]],
      dtype=object)
```



첫번째 모델 만들기

- knn model (k- nearest neighbor)

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=1)
knn
```

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

```
# 학습
knn.fit(X_train, Y_train)

print(type(X_train))  # numpy.ndarray

# 예측
X_new = np.array([[5, 2.9, 1, 0.2]])  # 똑같이 numpy.ndarray 형태로

pred = knn.predict(X_new)
pred
```

```
<class 'numpy.ndarray'>
array([0])
```

```
pred_targetname = iris['target_names'][pred]
pred_targetname
```

```
array(['setosa'], dtype='<U10')
```

```
# 평가하기
print("테스트 셋의 정확도 : {:.2f}".format(np.mean(pred == Y_test)))
#print("테스트 셋의 정확도 : {%.2f}")   # %형태
```

```
테스트 셋의 정확도 : 0.34
```

타이타닉 셋으로 실습

```
titanic = pd.read_csv('./data/train.csv')
test = pd.read_csv('./data/test.csv')
```

```
titanic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs	female	38.0	1	0	PC 17599	71.2833	C8

```
test.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S

```
pd.isnull(titanic).head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	False	False	False	False	False	False	False	False	False	False	True	Fals
1	False	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	True	Fals
3	False	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	True	Fals

```
titanic.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
titanic.Age.mean()
```

```
29.69911764705882
```

```
titanic.Age.fillna(30).head()
```

```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

```
titanic.Embarked.fillna(method='ffill').head()
```

```
0    S
1    C
2    S
3    S
4    S
Name: Embarked, dtype: object
```

```
titanic.Sex = pd.get_dummies(titanic.Sex)
```

```
titanic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs	1	38.0	1	0	PC 17599	71.2833	C85

```
titanic['Survived'] = titanic['Survived'].astype('category')
```

```
test.Sex = pd.get_dummies(test.Sex)
```

```
test.Age.fillna(30).head()
```

```
0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
Name: Age, dtype: float64
```

```
test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
## k=5로 설정
knn = KNeighborsClassifier(n_neighbors=3)
knn
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='uniform')
```

```
from sklearn.model_selection import train_test_split
f_names = ['Pclass','SibSp','Fare','Sex']
X_train, X_test, Y_train, Y_test = train_test_split(titanic[f_names],
                                                    titanic['Survived'],
                                                    random_state = 0)
```

```
knn.fit(X_train, Y_train)

#print(type(X_train))  # numpy.ndarray
#X_new = test[f_names]
# 예측
pred = knn.predict(X_test)
pred
```

```
array([0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1,
       0, 1, 1], dtype=int64)
```

```
an = pd.read_csv('./data/gender_submission.csv')
```

```
print("테스트 셋의 정확도 : {:.2f}".format(np.mean(pred == Y_test)))
```

```
테스트 셋의 정확도 : 0.78
```