

# IPA 주관 인공지능센터 기본(fundamental) 과정

- GitHub link: [here](#)
- E-Mail: windkyle7@gmail.com

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as mso
%matplotlib inline
```

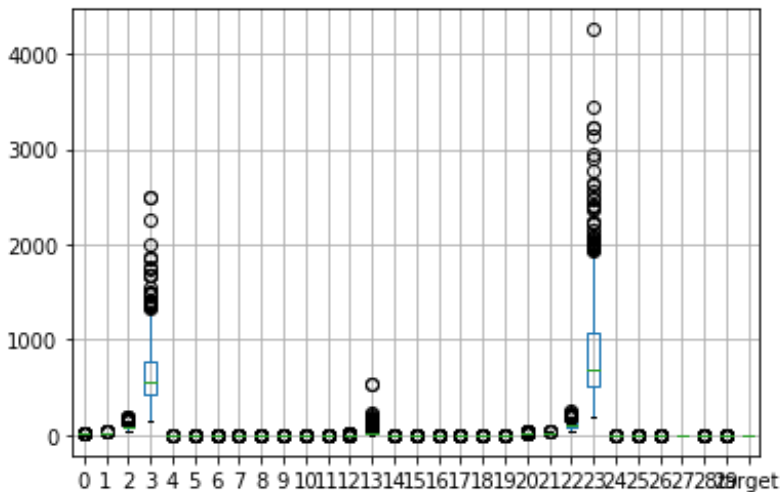
In [2]:

```
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()
X, y = pd.DataFrame(data.data), pd.DataFrame(data.target, columns=['target'])
cancer = pd.concat([X, y], axis=1)
cancer.boxplot()
```

Out[2]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c4423ab38>



In [3]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score

t = cross_val_score(KNeighborsClassifier(),
                    cancer.iloc[:, :-1],
                    cancer.iloc[:, -1],
                    cv=10)
print(np.mean(t), np.std(t))
```

0.9298429262812202 0.028921039975408073

In [4]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(cancer.iloc[:, :-1],
                                                    cancer.iloc[:, -1])
```

In [5]:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline

pipe = Pipeline([('scaler', MinMaxScaler()), ('knn', KNeighborsClassifier())])
pipe.fit(X_train, y_train)
```

Out[5]:

```
Pipeline(memory=None,
          steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
                 ('knn',
                  KNeighborsClassifier(algorithm='auto', leaf_size=30,
                                       metric='minkowski', metric_params=None,
                                       n_jobs=None, n_neighbors=5, p=2,
                                       weights='uniform'))],
          verbose=False)
```

In [6]:

```
pipe.steps
```

Out[6]:

```
[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
 ('knn',
  KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                      metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                      weights='uniform'))]
```

In [7]:

```
from sklearn.preprocessing import RobustScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import make_pipeline

make_pipeline(RobustScaler(), PCA())
```

Out[7]:

```
Pipeline(memory=None,
          steps=[('robustscaler',
                  RobustScaler(copy=True, quantile_range=(25.0, 75.0),
                               with_centering=True, with_scaling=True)),
                 ('pca',
                  PCA(copy=True, iterated_power='auto', n_components=None,
                      random_state=None, svd_solver='auto', tol=0.0,
                      whiten=False))],
          verbose=False)
```

In [8]:

```
from sklearn.preprocessing import StandardScaler
make_pipeline(StandardScaler(), PCA(n_components=2), StandardScaler())
```

Out[8]:

```
Pipeline(memory=None,
         steps=[('standardscaler-1',
                  StandardScaler(copy=True, with_mean=True, with_std=True)),
                 ('pca',
                  PCA(copy=True, iterated_power='auto', n_components=2,
                      random_state=None, svd_solver='auto', tol=0.0,
                      whiten=False)),
                 ('standardscaler-2',
                  StandardScaler(copy=True, with_mean=True, with_std=True))],
         verbose=False)
```

In [9] :

```
temp = []
for i in range(cancer.shape[1] - 1, 1, -1):
    pca = PCA(i)
    x = pca.fit_transform(cancer.iloc[:, :-1])
    knn = KNeighborsClassifier()
    t = cross_val_score(knn, x, cancer.iloc[:, -1], cv=10)
    temp.append(np.mean(t))
np.max(temp)
```

Out[9]:

0.9316286405669347

In [10]:

```
from sklearn.model_selection import GridSearchCV

for i in range(cancer.shape[1] - 1, 1, -1):
    pipe = make_pipeline(PCA(i), KNeighborsClassifier())
    grid = GridSearchCV(
        pipe, {'kneighborsclassifier__n_neighbors': [n for n in range(1, 11)]},
        cv=10,
        iid=True)
    grid.fit(cancer.iloc[:, :-1], cancer.iloc[:, -1])
    print(grid.best_params)
```

[illegible]

```
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 5}  
{'kneighborsclassifier__n_neighbors': 10}  
{'kneighborsclassifier__n_neighbors': 8}
```