

Machine Learning Operation(MLOps)

– Ch3. Level1 MLOps(2)

sklearn.pipeline

일반 코드

```
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA
```

```
sd = StandardScaler()
```

① `sd.fit(train)`

② `new_train = sd.transform(train)`

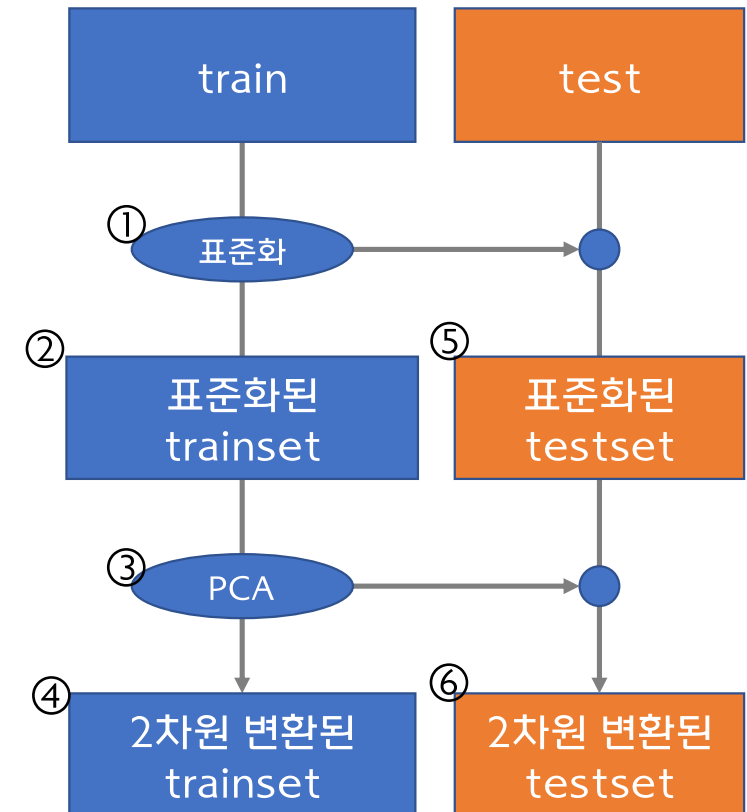
```
model = PCA(n_components=2)
```

③ `model.fit(new_train)`

④ `train_pca = model.transform(new_train)`

⑤ `new_test = sd.transform(test)`

⑥ `test_pca = model.transform(new_test)`

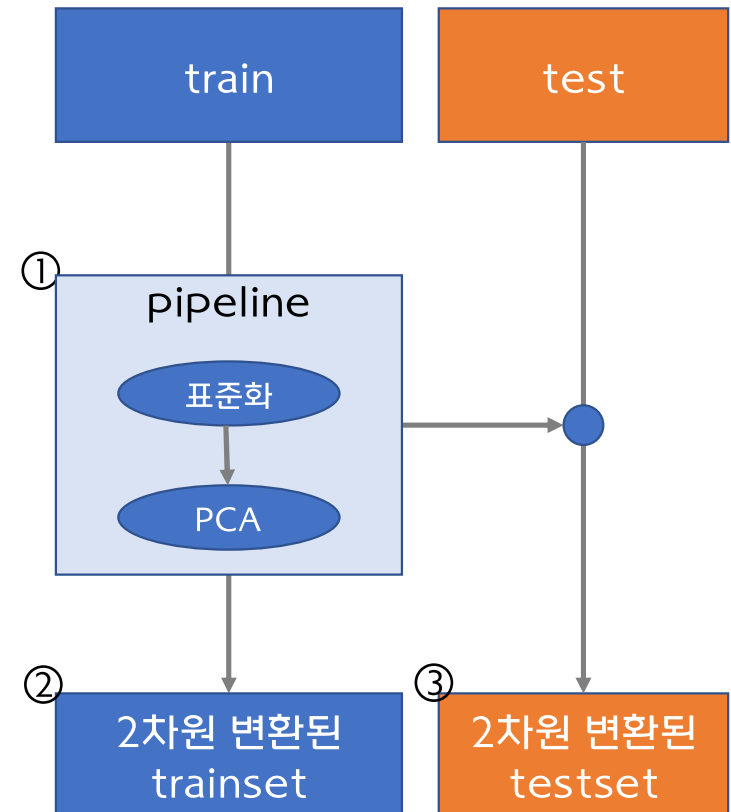


sklearn.pipeline 코드

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
```

```
pipe = Pipeline(steps = [
    ('step1', StandardScaler()),
    ('step2', PCA(n_components=2)),
])
```

- ① pipe.fit(train)
- ② train_pca = pipe.transform(train)
- ③ test_pca = pipe.transform(test)



sklearn.compose

일반 코드

```
from sklearn.preprocessing import OneHotEncoder, StandardScaler
```

```
ycol = ["Survived"]
```

```
xcols = [col for col in train.columns if col not in ycol]
```

```
① num_cols = train[xcols].select_dtypes(include=np.number).columns.tolist()  
   cat_cols = [col for col in xcols if col not in num_cols+ycol+["Name","Ticket","Cabin"]]
```

```
   cat_enc = OneHotEncoder(sparse_output=False)
```

```
② cat_enc.fit(train[cat_cols])
```

```
③ train_cat_encoded = cat_enc.transform(train[cat_cols])
```

```
   num_enc = StandardScaler()
```

```
④ num_enc.fit(train[num_cols])
```

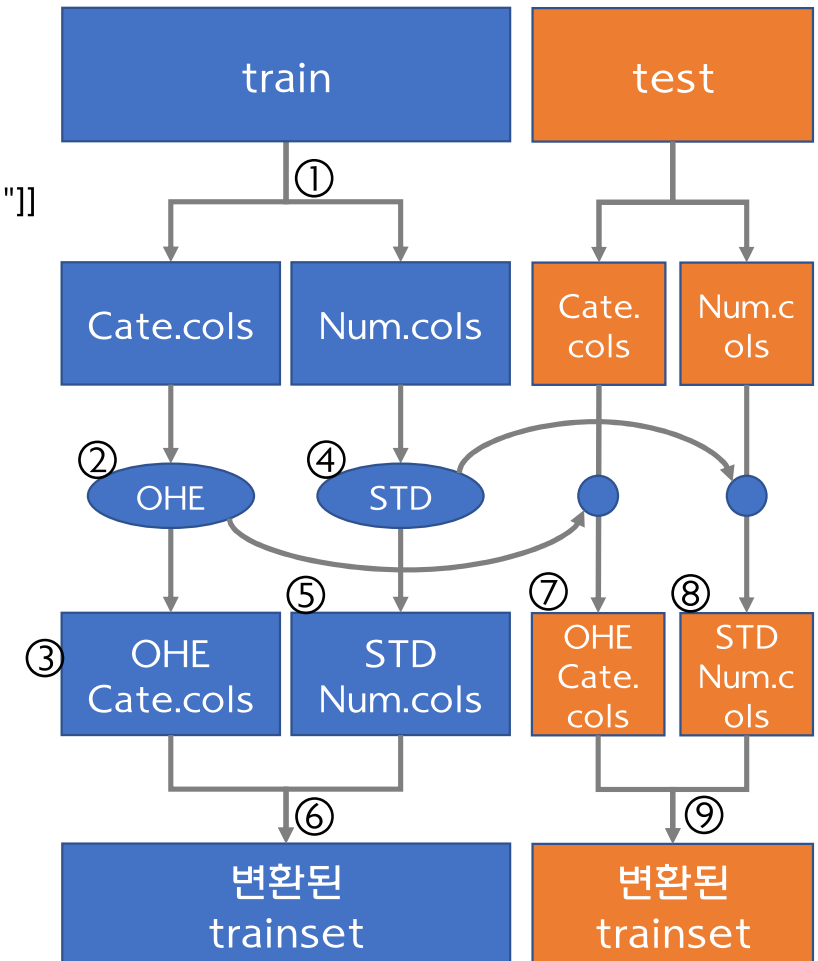
```
⑤ train_num_encoded = num_enc.transform(train[num_cols])
```

```
⑥ train_encoded = np.concatenate([train_cat_encoded, train_num_encoded], axis=1)
```

```
⑦ test_cat_encoded = cat_enc.transform(test[cat_cols])
```

```
⑧ test_num_encoded = num_enc.transform(test[num_cols])
```

```
⑨ test_encoded = np.concatenate([test_cat_encoded, test_num_encoded], axis=1)
```



sklearn.compose 코드 - ColumnTransformer

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
```

```
ycol = ["Survived"]
xcols = [col for col in train.columns if col not in ycol]
```

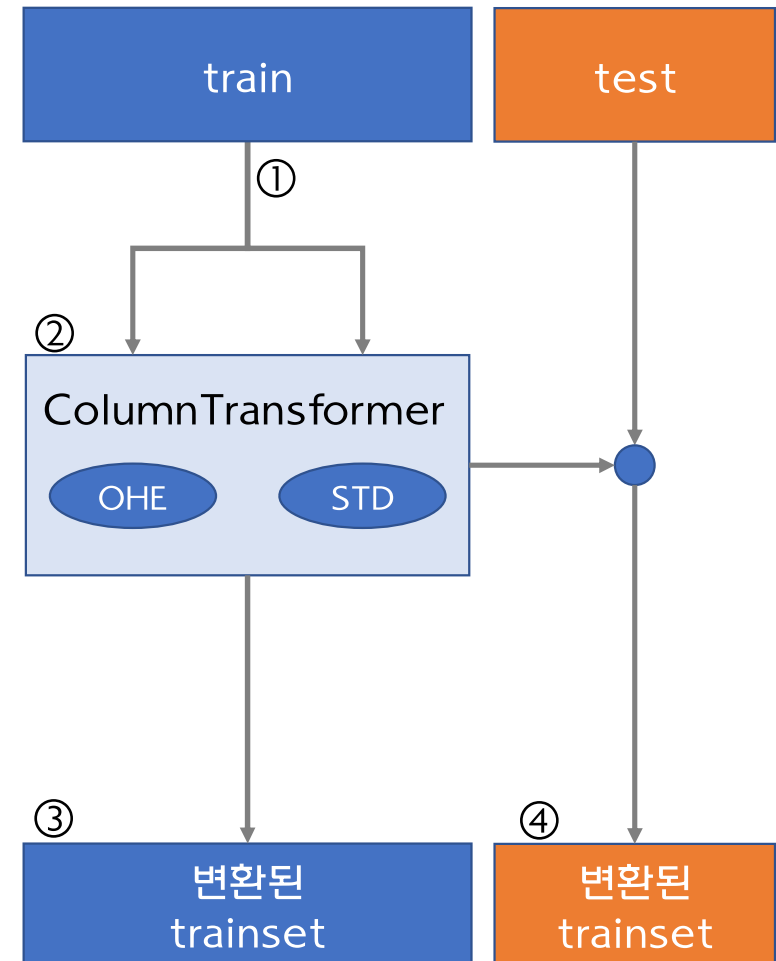
```
① num_cols = train[xcols].select_dtypes(include=np.number).columns.tolist()
   cat_cols = [col for col in xcols if col not in num_cols+ycol+["Name","Ticket","Cabin"]]
```

```
trans = ColumnTransformer([
    ('cat_cols',OneHotEncoder(sparse_output=False), cat_cols),
    ('num_cols',StandardScaler(), num_cols)])
```

```
② trans.fit(train[xcols])
```

```
③ train_encoded = trans.transform(train[xcols])
```

```
④ test_encoded = trans.transform(test[xcols])
```



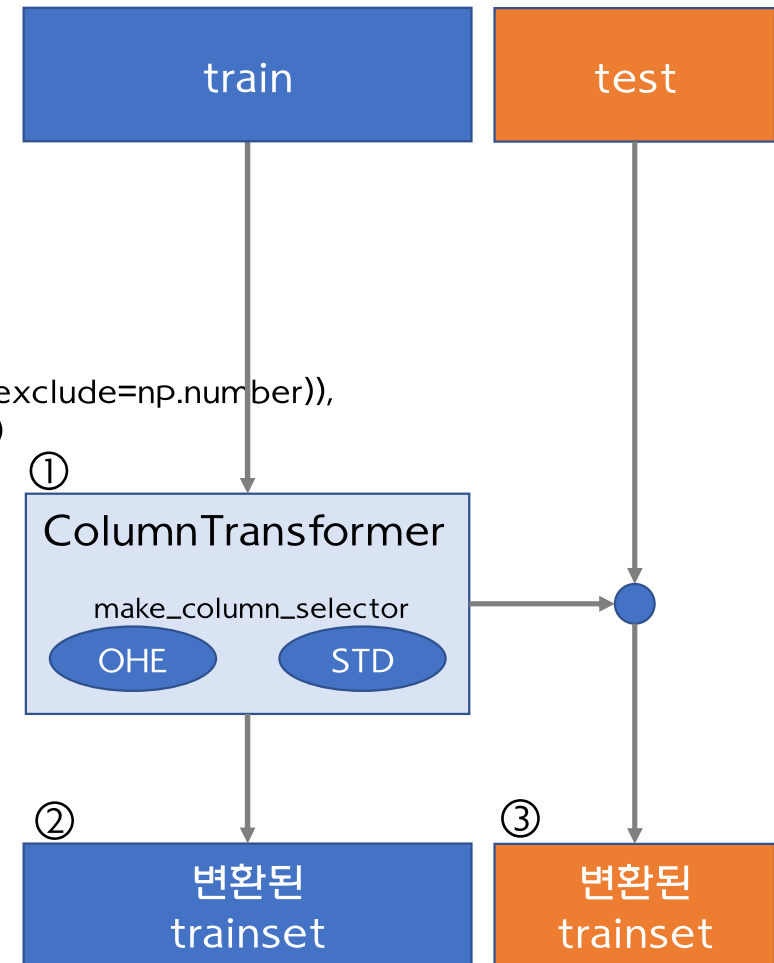
sklearn.compose 코드 – make_column_selector

```
from sklearn.compose import ColumnTransformer, make_column_selector
from sklearn.preprocessing import OneHotEncoder, StandardScaler
```

```
ycol = ["Survived"]
xcols = [col for col in train.columns if col not in ycol+["Name","Ticket","Cabin"]]
```

```
trans = ColumnTransformer([
    ('cat_cols',OneHotEncoder(sparse_output=False), make_column_selector(dtype_exclude=np.number)),
    ('num_cols',StandardScaler(), make_column_selector(dtype_include=np.number))])
```

- ① trans.fit(train[xcols])
- ② train_encoded = trans.transform(train[xcols])
- ③ test_encoded = trans.transform(test[xcols])



End of Document