

ESC

Scikit-learn Pipeline 사용법

Scikit-learn Pipeline 사용법

Scikit-learn Pipeline 사용법에 대해 알아보시다.

1

What?

Pipeline은 뭐고
왜 써야하나??

2

How?

그래서 어떻게 쓰는건데??

3

Example

예시

What?

우선 정의를 알아보자.

사이트에 따르면...

"The sklearn.pipeline module implements utilities to build a composite estimator, as a chain of transforms and estimators."

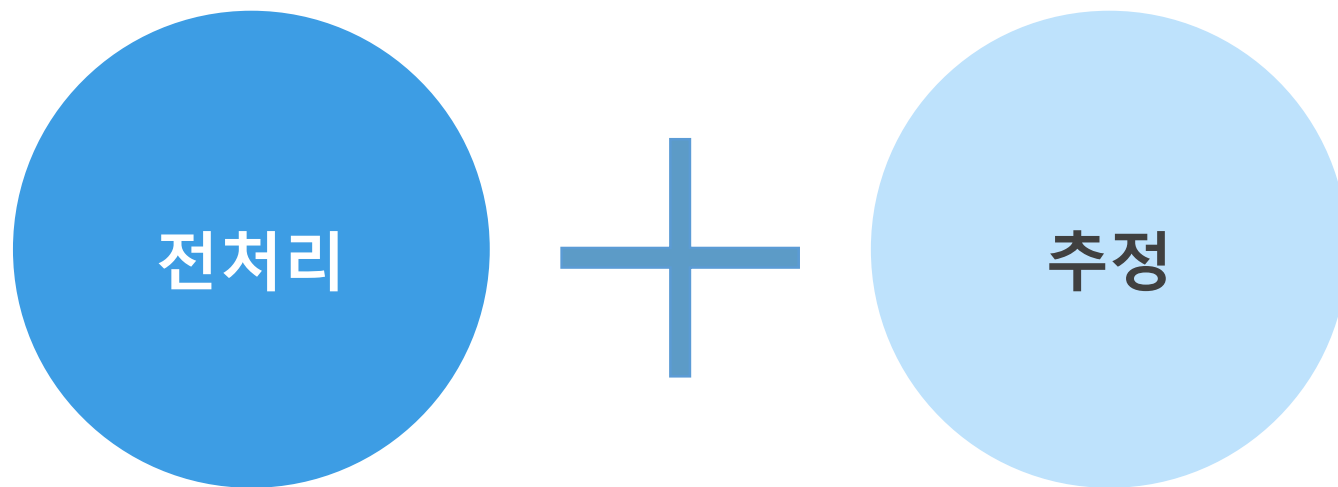


봐도 감이 잘 안 온다.

What?

정확한 정의는 연속된 변환을 순서대로 처리할 수 있게 해주는 Module이지만..

그냥 쉽게 얘기해서!



원래는 따로 처리했던 전처리과정과 추정과정을 하나로 합쳐주는 고마운 Module이라고 보면 된다.

즉, 데이터를 변환도 해주고 추정도 해주는 일석이조의 효과를 얻을 수 있다!

또한 여러 전처리 과정이 있는 경우에 한 번에 묶어서 순서대로 처리할 수 있다.

How?

본격적인 사용방법에 앞서 변환기(Transformer)와 추정기(Estimator)에 대해 알아보자.

1 변환기(Transformer)

어떠한 데이터를 다른 데이터 형태로 변환시켜주는 애들을 말한다.

변환기들은 반드시 `<fit_transform>` method를 가져야한다.

Ex) PCA, SVC, Binarizer, StandardScaler, etc.

2 추정기(Estimator)

데이터들을 가지고 어떤 값들을 추정할 수 있도록 해주는 애들이다.

추정기는 분류기(Classifier)처럼 타겟이 되는 값들을 추정해주는 애들일 수도 있고,

앞서 말했던 변환기들도 추정기에 들어갈 수 있다.

Ex) 변환기들, MultinomialNB, LinearRegression, etc.

How?

* Pipeline은 여러 개의 이름/추정기 쌍을 목록으로 입력 받아서 순서대로 처리해주는 Module.

(이때 이름은 추정기의 이름은 우리가 지정해 줄 수 있고 make_pipeline을 통해 자동으로 지정해줄 수도 있다.)

* 그런데 주의해야할 점은 pipeline에는 여러 개의 추정기를 넣을 수 있지만, 마지막을 제외한 추정기들은 모두 변환기여야 한다.

(왜냐하면 중간에 변환기가 아닌 추정기가 들어가버리면 그 추정기에서 나온 결과값을 다음 단계에서 사용할 수 없는 상황이 발생하기 때문.)

Ex) pipeline = Pipeline([('scaler', StandardScaler()), ('clf', LogisticRegression())])

↓
변환기

↓
추정기

How?

1 Pipeline 만들기

앞 장 예시에서 간단하게 봤듯이 pipeline을 만드는 방법은 매우 간단하다!

Ex) Module 이름

```
from sklearn.pipeline import Pipeline
```

```
from sklearn.linear_model import LinearRegression
```

```
model = Pipeline([  
    ('scaler', StandardScaler()),  
    ('regressor', LinearRegression()),  
])
```

우리가 지정해줄 '이름'을 먼저 써주고 추정기의 원래 이름을 다음에 적는다.
중요한 건 추정기의 parameter들을 이 때 지정해줄 수 있다.

How?

Pipeline의 Method

pipeline으로 결합된 모형은 원래의 모형이 가지는 <fit>과 <predict> method를 가진다.

각 method가 호출되면 그에 따른 적절한 method를 pipeline의 각 객체에 대해서 호출한다.

Fit

Pipeline에 대해 fit method를 호출하면 전처리 객체에는 fit_transform이 내부적으로 호출되고
분류 모형에서는 fit 메서드가 호출된다.

Predict

Pipeline에 대해 predict method를 호출하면 전처리 객체에는 transform이 내부적으로 호출되고
분류 모형에서는 predict method가 호출된다.

How?

2 Feature Union

문제 상황

수치형 변수와 범주형 변수는 변환하는 방법이 다르므로 pipeline을 각각 만들었다.

그런데 나는 각각의 파이프라인을 하나로 합쳐서 한 번에 일을 처리하고 싶다. 가능할까??

Feature Union을 쓰자!

예
시

숫자형 변수를 전처리하는 Pipeline

```
num_pipeline = Pipeline([
    ('selector', DataFrameSelector(num_attr)),
    ('imputer', Imputer(strategy = 'median')),
    ('std_scaler', StandardScaler())
])
```

범주형 변수를 전처리하는 Pipeline

```
cat_pipeline = Pipeline([
    ('selector', DataFrameSelector(cat_attr)),
    ('cat_encoder', CategoricalEncoder(encoding = 'onehot-dense'))
])
```



num_pipeline과 cat_pipeline을 합치는 FeatureUnion

```
full_pipeline = FeatureUnion(transformer_list = [
    ('num_pipeline', num_pipeline),
    ('cat_pipeline', cat_pipeline),
])
```

전체 파이프라인 실행

```
housing_prepared = full_pipeline.fit_transform(housing)
print(housing_prepared)
```

Example

3 Example

Pipeline을 사용하면 교차검증(Cross Validation)을 더 정확하고 수월하게 진행할 수 있다.

코드 예시 * 목표: SVM을 통해 글의 종류를 분류하는 모델을 만들고 교차검증도 해보자.

```
From sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer
From sklearn.metrics import f1_score
From sklearn.model_selection import cross_val_score
From sklearn.svm import LinearSVC
From sklearn.pipeline import Pipeline
From sklearn.datasets import fetch_20newsgroups
```

필요한 Module
불러오기

Example

```
cats = ['alt.atheism', 'sci.space']  
newsgroups_train = fetch_20newsgroups(subset='train', categories=cats)  
newsgroups_test = fetch_20newsgroups(subset='test', categories=cats)
```

```
X_train = newsgroups_train.data  
X_test = newsgroups_test.data  
y_train = newsgroups_train.target  
y_test = newsgroups_test.target
```

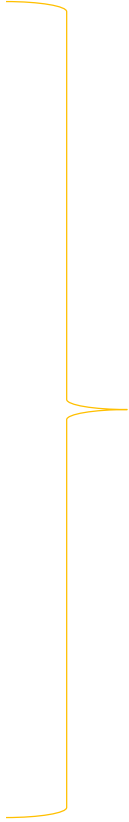
Train/Test Data Set
분할

Example

```
# this calculates a vector of term frequencies for  
# each document  
vect = CountVectorizer()
```

```
# this normalizes each term frequency by the  
# number of documents having that term  
tfidf = TfidfTransformer()
```

```
# this is a linear SVM classifier  
clf = LinearSVC()
```



변환기와 추정기 설정

Example

```
pipeline = Pipeline([  
    ('vect',vect),  
    ('tfidf',tfidf),  
    ('clf',clf)  
])
```

Pipeline 만들기

```
scores = cross_val_score(pipeline ,X_train ,y_train ,cv=3 ,scoring='f1_micro')
```

```
print(scores)
```


```
print(scores.mean())
```

Cross-validation
Score 체크

Example

```
# now train and predict test instances
pipeline.fit(X_train,y_train)
y_preds = pipeline.predict(X_test)

# calculate f1
f1_score(y_test, y_preds, average='micro')
```



Pipeline fit을 이용해서
y값을 예측한 후
f1 score 구하기

참고자료

1. <https://rk1993.tistory.com/entry/Python-sklearnpipeline-%ED%8C%8C%EC%9D%B4%ED%94%84%EB%9D%BC%EC%9D%B8Pipeline%EC%9D%B4%EB%9E%80>
2. <https://stickie.tistory.com/77?category=790728>
3. <https://data-newbie.tistory.com/32>
4. <https://datascienceschool.net/view-notebook/f43be7d6515b48c0beb909826993c856/>