

16

오버 샘플링 기법

[수업 목표]

이번 시간에는 오버 샘플링 기법에 대해 배워보겠습니다

[수업 개요]

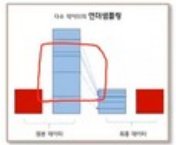
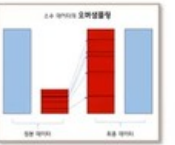
00:09 오버 샘플링이란?

jupyter 16.2_오버샘플링 (autosaved)

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (pykernel)


16.2 오버 샘플링

- 소수 클래스의 샘플을 증가시키거나 다이나믹 클래스와 메이저 클래스의 샘플 크기를 동일하게 만드는 기법
- 컨덤 오버샘플링(소수 샘플 복제)은 동일한 정보를 복사하여 오버피팅을 유발할 수 있음

16.2.1 SMOTE(Synthetic Minority Over-sampling Technique)

- 소수 클래스의 기존 샘플을 사용하여 새로운 합성 군집을 생성하는 오버 샘플링 기법. 소수 클래스에 대한 선형 보간법으로 가장 흔한 기록을 생성.
- 합성 훈련 기록은 소수 클래스의 각 예에 대해 k -최근접 이웃 중 하나 이상을 무작위로 선택하여 생성.
- 오버샘플링 과정을 거친 후 데이터를 재구성하고 처리된 데이터에 대해 여러 분류 모델을 적용할 수 있음
- 데이터의 특성에 따라 다르겠지만, 복제된 데이터를 분석을 위해서는 많은 데이터 확보가 효과적이므로 오버샘플링 기법을 적용하는 것이 좋음.
- 기존의 데이터가 적은 새로운 사례의 데이터에서 사용하기 어려움.





소수 데이터 클래스의 보강

01:05 SMOTE


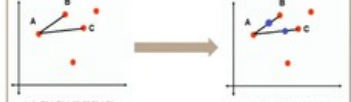
jupyter 16.2_오버샘플링 (autosaved)

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (pykernel)

16.2.1 SMOTE(Synthetic Minority Over-sampling Technique)

- 소수 클래스의 기존 샘플을 사용하여 새로운 합성 군집을 생성하는 오버 샘플링 기법. 소수 클래스에 대한 선형 보간법으로 가장 흔한 기록을 생성.
- 합성 훈련 기록은 소수 클래스의 각 예에 대해 k -최근접 이웃 중 하나 이상을 무작위로 선택하여 생성.
- 오버샘플링 과정을 거친 후 데이터를 재구성하고 처리된 데이터에 대해 여러 분류 모델을 적용할 수 있음
- 데이터의 특성에 따라 다르겠지만, 복제된 데이터를 분석을 위해서는 많은 데이터 확보가 효과적이므로 오버샘플링 기법을 적용하는 것이 좋음.
- 기존의 데이터가 적은 새로운 사례의 데이터에서 사용하기 어려움.

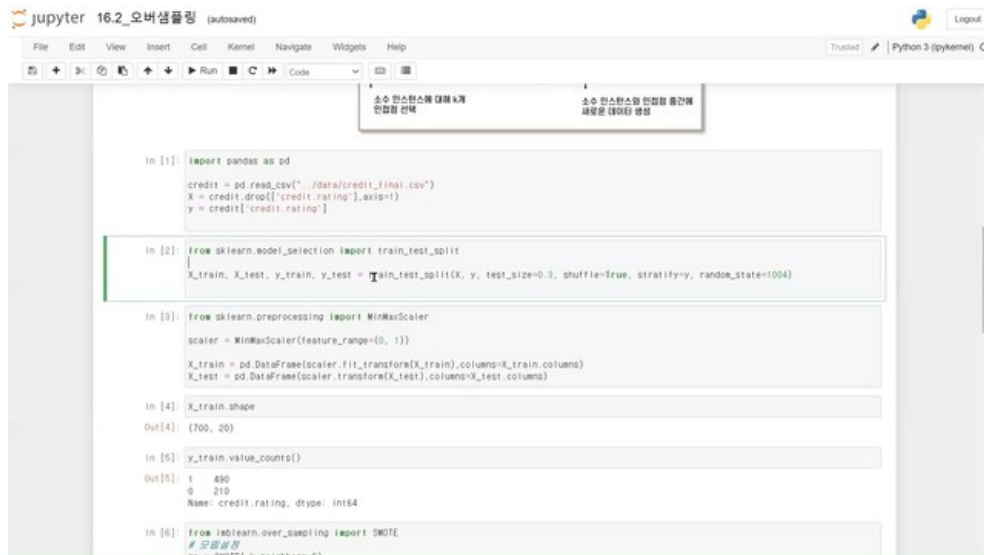



소수 데이터 클래스의 보강

소수 클래스에 대해 k 개 인접한 선택

소수 클래스를 인접한 중간에 새로운 데이터 생성

03:16 실행



```

In [1]: import pandas as pd

credit = pd.read_csv("../data/credit_final.csv")
X = credit.drop(['credit_rating'], axis=1)
y = credit['credit_rating']

In [2]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, shuffle=True, stratify=y, random_state=1004)

In [3]: from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
X_train = pd.DataFrame(scaler.fit_transform(X_train), columns=X_train.columns)
X_test = pd.DataFrame(scaler.transform(X_test), columns=X_test.columns)

In [4]: X_train.shape
Out[4]: (700, 20)

In [5]: y_train.value_counts()
Out[5]: 1    490
        0    210
        Name: credit_rating, dtype: int64

In [6]: from sklearn.over_sampling import SMOTE
# 데이터 증강
sm = SMOTE(k=5, random_state=1)
  
```

[다음 수업 예고]

다음 시간에는 데이터 처리 실습에 대해 배워보겠습니다