머신러닝

Week-06. 모델 성능 향상

Jungwon Seo, 2021-Spring

목차

1. 머신러닝 파이프라인을 만들때 주의 할점

데이터 분리 단계
정보 누설

- 정보누설이란, 모델 구축 과정에서 검증/테스트 데이터의 정보가 이미 반영되어 있는 경우
  - 낙관적인 결과를 만들 수 있다.
  - 정보누설을 피하기 위해서는 전처리에 앞서 데이터 분리과정을 미리 진행을 하여야 한다.
  - 그 후에, 전처리와 (스케일링, 인코딩) 등을 훈련 데이터 셋에 대해서만 진행 한다.

```
scaled_cancer_data = MinMaxScaler().fit_transform(cancer.data)

X_train, X_test, y_train, y_test = train_test_split(
scaled_cancer_data, cancer.target, random_state=0)

svm = SVC(gamma='auto')
svm.fit(X_train, y_train)
print("테스트 점수: {:.2f}".format(svm.score(X_test, y_test)))
```

```
테스트 점수: 0.96
```

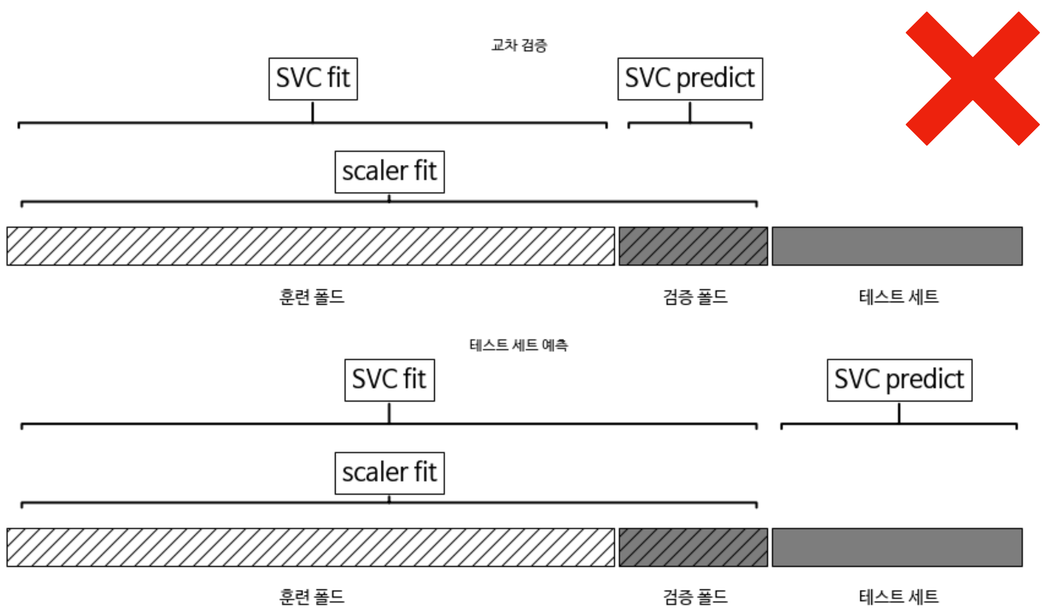
```
# 데이터 분할
X_train, X_test, y_train, y_test = train_test_split(
    cancer.data, cancer.target, random_state=0)

# 훈련 데이터의 최솟값, 최댓값을 계산합니다
scaler = MinMaxScaler().fit(X_train)

# 훈련 데이터의 스케일을 조정합니다
X_train_scaled = scaler.transform(X_train)

svm = SVC(gamma='auto')
# 스케일 조정된 훈련데이터에 SVM을 학습시킵니다
svm.fit(X_train_scaled, y_train)
# 테스트 데이터의 스케일을 조정하고 점수를 계산합니다
X_test_scaled = scaler.transform(X_test)
print("테스트 점수: {:.2f}".format(svm.score(X_test_scaled, y_test)))
```

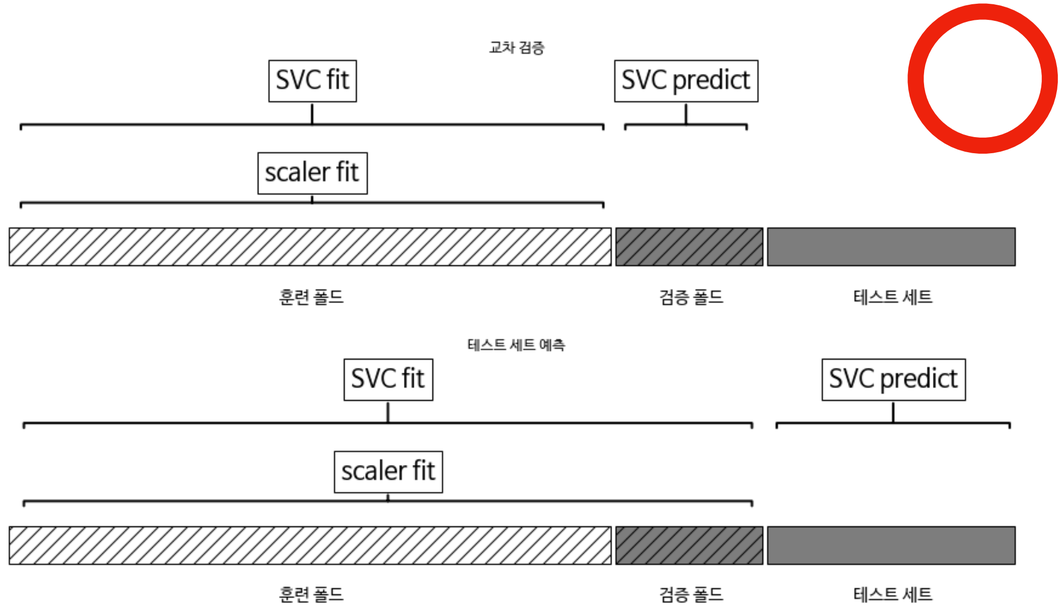
```
테스트 점수: 0.95
```

데이터 분리 단계

교차 검증에서의 정보누설

- 교차 검증에서도 훈련 폴드와 검증 폴드를 고립 시킨 상태에서 훈련 폴드만을 기반으로 스케일러나 인코더를 생성해야함





2. 영혼까지 끌어 올려 성능 올리기

성능 올리기
데이터 관점 - 인코딩

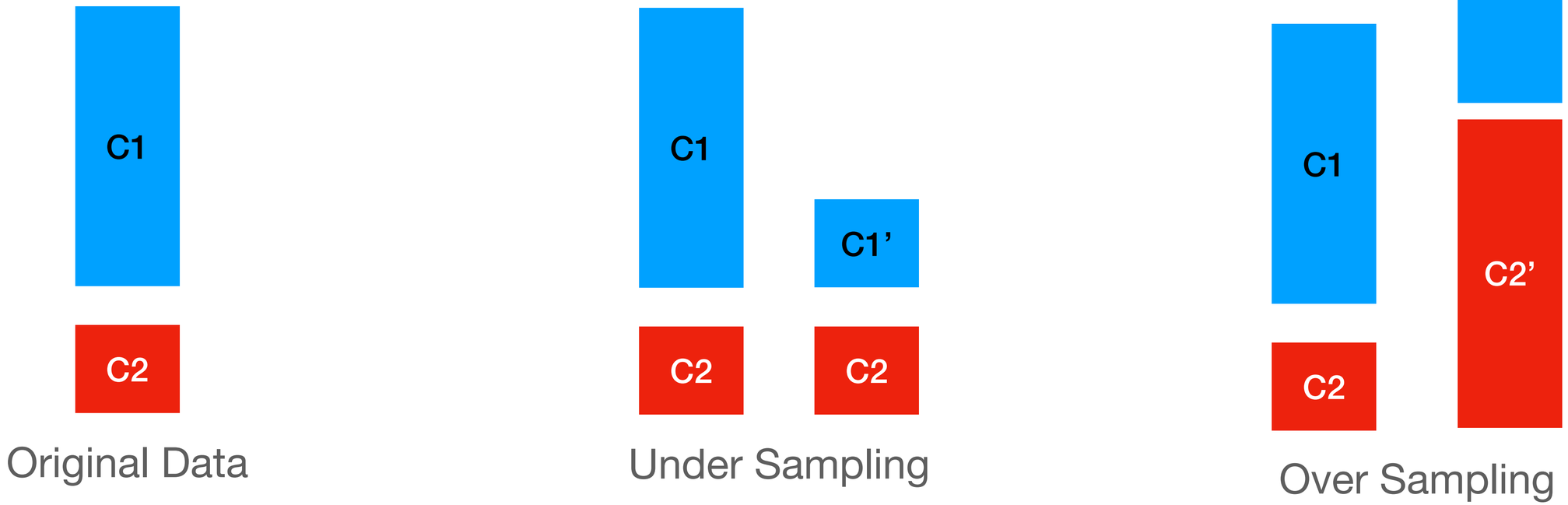
- 데이터 인코딩 및 Binning
- 범주형 데이터 인코딩
  - Long Tail 분포의 데이터의 경우
  - One-hot 인코딩의 경우 전체 특성 값들의 빈도수를 확인 한 후에 상위 N개 + 기타 형태로 재구성 해에 인코딩
- 연속형 데이터 Binning
  - 연속형 데이터를 특정 구간으로 나누어서 범주형 데이터로 변경
  - 아웃라이어에 의한 영향 최소화
  - 예) 재산을 0원부터 100억까지 연속형 데이터로 가져가지않고, [0~5천만원], [5천만원~1억], [1억~10억], [10억 이상]

성능 올리기
데이터 관점 - 스케일링

- 특성 Scaling
- 서로 다른 특성간의 단위 차이가 너무 심할 경우 일부 모델에서는 변수의 영향이 제대로 반영되지 않음
  - 재산 특성: 0원에서 100,000,000까지
    차량 보유 수: 0대에서 4대까지
  - 직관적으로 봤을 때 차량 1대가 더 많은 건 엄청난 차이지만, 재산 기준으로 보면 1원과 크게 다르지 않음
  - MinMax 스케일링 적용
    - 재산: 0원, 50만원, 100만원, 1천만원, 1억 => [0.0, 0.005, 0.01, 0.1 ,1.0]
    - 차량: 0대, 1대, 2대, 3대, 4대 => [0.0, 0.25, 0.5, 0.75, 1.0]
  - KNN 알고리즘을 생각해보자!

성능 올리기
데이터 관점

- 클래스 불균형 (Class Imbalance) 데이터
  - 클래스 불균형 데이터의 경우 샘플링을 다르게 하여 데이터를 재 구성 할 수 있습니다.
  - 이때 단순히 샘플링만 다르게 하면 효과가 없고, 앙상블을 적용해야 효과가 있습니다.

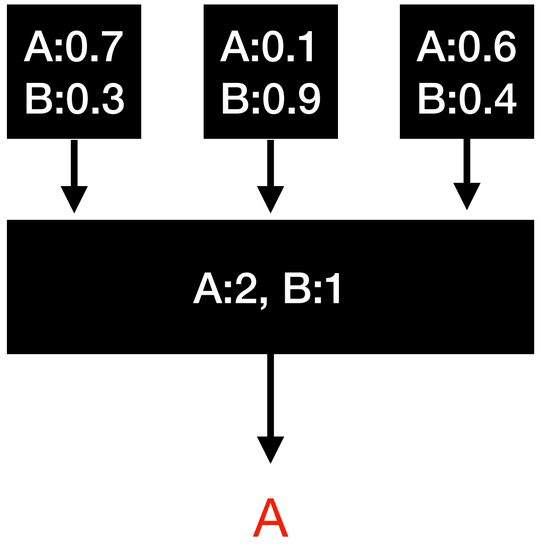


성능 올리기
모델 관점: 앙상블

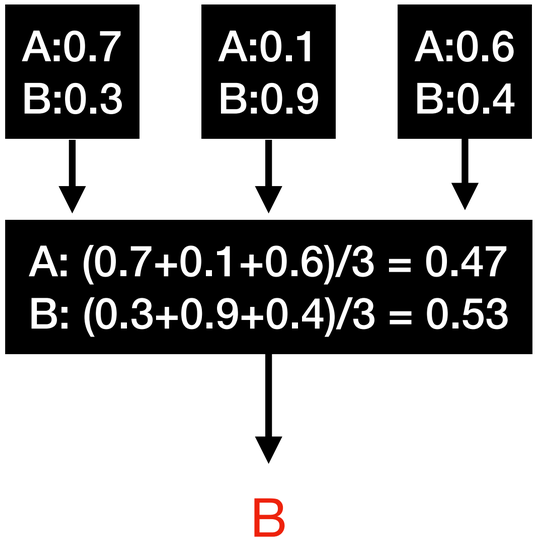
- 앙상블
  - 서로다른 환경 (데이터 분포, 전처리, 모델) 를 구성하여 학습된 모델의 조합으로 일반화된 최종 모델을 생성
- Hard Voting: Majority Voting
  - 다수결의 원칙에 따라 최종 클래스를 결정
- Soft Voting: Probability Voting
  - 각각의 모델에서 나온 확률값의 평균으로 최종 클래스를 결정
- Weighted Voting
  - 특정 모델에 가중치를 주어서 최종 클래스를 결정

성능 올리기

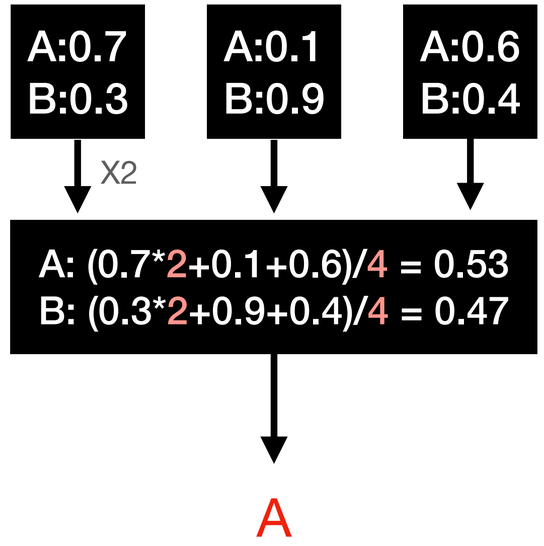
모델 관점: 앙상블



Hard Voting



Soft Voting



Weighted Voting

E.O.D