11. 데이터 전처리와 파생변수 생성

· 데이터 분석이나 머신러닝 모델링의 효과를 제대로 얻기 위해서는 데이터 전처리 과정이 매우 중요하다.
· 잘못된 데이터는 통계적 편향을 증가시키므로, 의미 있는 인사이트를 얻기 힘들다.

11.1 결측값 처리

· 실제 분석 프로젝트에서 다루는 대부분의 데이터는 결측값(missing value)이나 이상치가 없는 경우가 드물기 때문에, 데이터 탐색 단계에서 파악한 문제점들을 처리하는 과정이 필요하다.
· 가공되지 않은 데이터는 상당량의 결함을 가지고 있고, 결측값 처리방법을 결정하기 전에 데이터 탐색을 통해 결측값의 비율이 어떻게 되는지, 한변수에 결측값이 몰려 있지는 않은지 등을 파악해야 한다.
(* 어떤 경우에는 빈 문자열이 입력되어 있어 결측값으로 인식되지 않을 수도 있으므로 확인 필요!)
· 결측값은 분석 환경에 따라 '.', 'NA', 'NaN' 등으로 표시한다.

완전 무작위 결측(MCAR: Missing Completely at Random)

- 순수하게 결측값이 무작위로 발생한 경우
- 결측값을 포함한 데이터를 제거해도 편향(bias)이 거의 발생되지 않는다.

무작위 결측(MAR: Missing at Random)

- 다른 변수의 특성에 의해 해당 변수의 결측치가 체계적으로 발생한 경우

비무작위 결측(NMAR: Missing at Not Random)

- 결측값들이 해당 변수 자체의 특성을 갖는 경우

결측값 처리 방법

표본 제거 방법(Completes analysis)

: 가장 간단한 결측값 처리 방법으로, 결측값이 심하게 많은 변수를 제거하거나 결측값이 포함된 행(observations)을 제외하고 데이터 분석을 하는 표본 제거 방법(Completes analysis).

- 전체 데이터에서 결측값 비율이 10% 미만일 경우 사용한다.

평균 대치법(Mean Imputation)

: 결측값을 제외한 온전한 값들의 평균을 구한 다음, 그 평균 값을 결측값들에 대치한다.

장점)

사용하기 간단하고 결측 표본 제거 방법의 단점을 어느 정도 보완 가능하다.

단점)

관측된 데이터의 평균을 사용하기 때문에 통계량의 표준오차가 왜곡되어 축소되어 나타나고, p-value가 부정확하게 된다.

보간법(interpolation)

- 시점 인덱스의 간격이 불규칙하거나 결측값이 두 번 이상 연달아 있을 때는 선형적인 수치 값을 계산
- 우선 데이터를 시간 순으로 정렬

회귀 대치법(regression imputation)

: 추정하고자 하는 결측값을 가진 변수를 종속변수로 하고, 나머지 변수를 독립 변수로 하여 추정한 회귀식을 통해 결측값을 대치

단점)

결측된 변수의 분산을 과소 추정할 가능성이 있다.

확률적 회귀 대치법(stochastic regression imputation)

- 회귀 대치법의 문제를 해결하기 위해 인위적으로 회귀식에 확률 오차항을 추가하여 변동성 조정
- 관측된 값들을 변동성만큼 결측값에도 같은 변동성을 추가

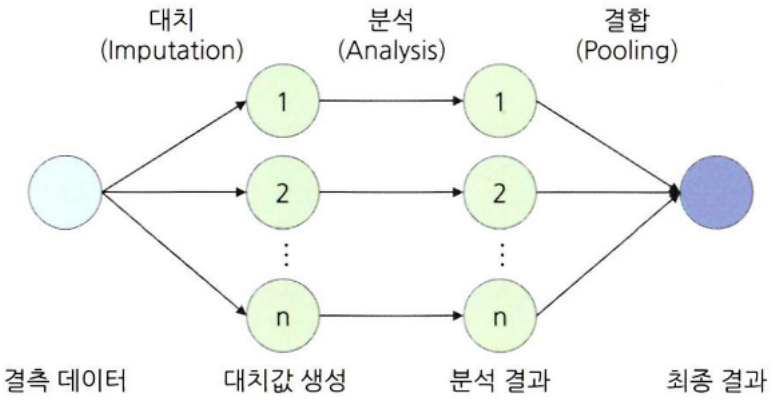
단점)

여전히 어느 정도 표본오차를 과소 추정한다.

다중 대치법(multiple imputation)

- 단순대치를 여러 번 수행하여 n개의 가상적 데이터를 생성하여 이들의 평균으로 결측값을 대치한다.

- 대치 단계(Imputations step) : 가능한 대치 값의 분포에서 추출된 서로 다른 값으로 결측치를 처리한 n개의 데이터셋 생성
- 분석 단계(Analysis step): 생성된 각각의 데이터셋을 분석하여 모수의 추정치와 표준오차 계산
- 결합 단계(Pooling step): 계산된 각 데이터셋의 추정치와 표준오차를 결합하여 최종 결측 대치값 산출



11.2 이상치 처리

이상치(outlier)
: 일부 관측치의 값이 전체 데이터의 범위에서 크게 벗어난 아주 작거나 큰 극단적인 값을 갖는 것.

- 데이터의 모집단 평균이나 총합 추정에 문제를 일으키며, 분산을 과도하게 증가시켜 분석이나 모델링의 정확도를 감소시키기 때문에 제거하는 것이 좋다.
- 전체 데이터의 양이 많을수록 튀는 값이 통곗값에 미치는 영향력이 줄어들어 이상치 제거의 필요성이 낮아진다.

해당 값을 결측값으로 대체한 다음 결측값 처리를 하거나, 아예 해당 이상치를 제거(trimming)

단점)
추정치의 분산은 감소하지만, 실젯값을 과장하여 편향을 발생시킨다.

따라서, 하한 값과 상한 값을 결정한 후 하한 값보다 작으면 하한 값으로 대체하고 상한 값보다 크면 상한 값으로 대체하는 관측값 변경(value modification)이나 이상치의 영향을 감소시키는 가중치를 주는 가중치 조정(weight modification) 방법을 사용한다.

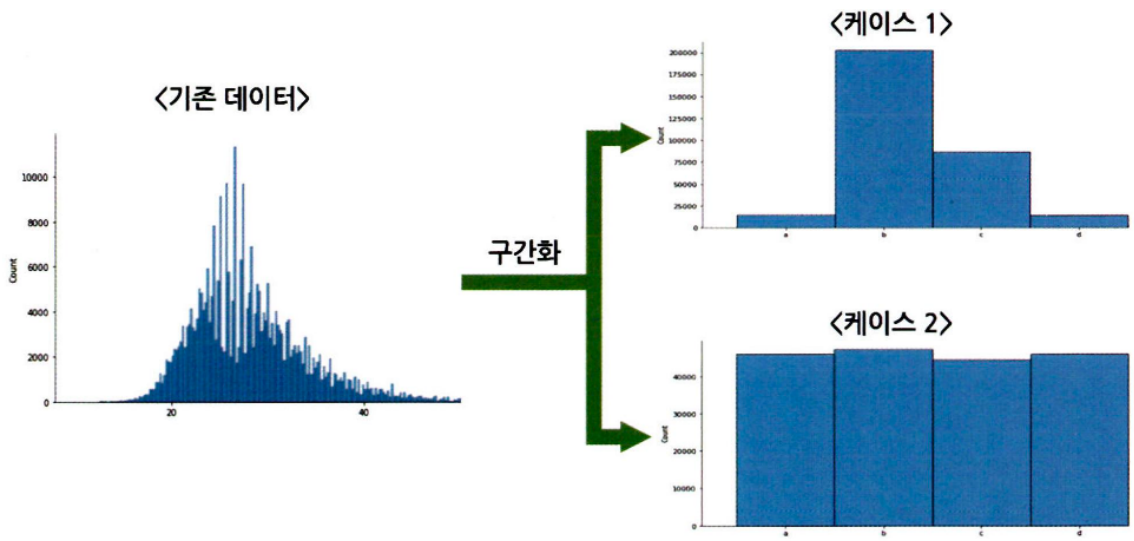
이상치의 선정법

- 박스플롯 상에서 분류된 극단치를 그대로 선정
- 임의로 허용범위를 설정하여 이를 벗어나는 자료를 이상치로 정의

평균(중위수)으로부터 ±n 표준편차 이상 떨어져 있는 값을 이상치로 보는데, 보통 n은 3으로 하지만 경우에 따라 설정. 분포가 비대칭인 데이터의 경우에는 -n과 +n 표준편차 값을 서로 다르게 설정하기도 한다. 좀 더 정교하게 들어가면, 평균은 이상치에 통계량이 민감하게 변하기 때문에, 이상치에 보다 강건한 중위수와 중위수 절대 편차(MAD: Median Absolute Deviation)를 사용하는 것이 좀 더 효과적이다.

11.3 변수 구간화(Binning)

데이터 분석의 성능을 향상시키기 위해, 혹은 해석의 편리성을 위해 이상형 변수를 범주형 변수로 변환한다.



위의 그림처럼 각 범주에 해당되는 관측치의 수가 유사해지도록 하여 범주별 분포가 일정하도록 구간화를 하는 방법도 사용.

⇨ 이산형 변수를 범주형 변수로 비즈니스적 상황에 맞도록 변환시킴으로써 데이터의 해석이나 예측, 분류 모델을 의도에 맞도록 유도 가능.

구간화하는 또 다른 방법들

- 동일 폭(width)으로 변수 구간화
- 동일 빈도(frequency)로 변수 구간화
- 구간별 평균 값으로 평활화(Smoothing)
- 구간별 중앙값으로 평활화
- 구간별 경곗값으로 평활화

변숫값이 효과적으로 구간화됐는지는 WOE(Weight of Evidence)값, IV(Information Value)값 등을 통해 측정 가능. IV 수치가 높을수록 종속변수의 True와 False를 잘 구분할 수 있는 정보량이 많다는 의미이므로, 변수가 종속변수를 제대로 설명할 수 있도록 구간화가 잘 되면 *IV값이 높아진다.*.

11.4 데이터 표준화와 정규화 스케일링

독립 변수들이 서로 단위가 다르거나 편차가 심할 때 값의 스케일을 일정한 수준으로 변환시켜주는 표준화(Standardization)와 정규화(Normalization) 스케일링한다.

해석적 관점에서, 데이터 표준화와 정규화는 매우 유용하다.
예를 들어 자동차 업체에서 고객의 항목별 소비금액에 따라 고객 세그먼테이션을 하고자 했을 때, 기호물품에 800 이상을 소비하고, 도서품목에 220 이하로 소비하는 고객은 A라는 차종을 추천하는 것이 적합하다는 결과가 나왔다고 가정한다. 이 경우 기호품목 800과 도서품목 220이 어느 정도 수준인지 바로 알아보기 어렵다. 그렇기 때문에 이 값이 평균보다 어느 정도 크거나 작은 지 바로 알 수 있는 수치로 변환시킴으로써 분석 내용을 효율적으로 해석할 수 있도록 하는 것이다.

표준화
각 관측치의 값이 전체 평균을 기준으로 어느 정도 떨어져 있는지 나타낼 때 사용
평균은 0으로 변환, 1표준편차 거리는 ±1, 2표준편차 거리는 ±2로 변환
Zero-mean으로부터 얼마나 떨어져 있는지를 나타내기 때문에 이를 Z-score라 표현
서로 다른 변수 간 값의 크기를 직관적으로 비교 불가
각 관측치 값에서 평균을 빼 준 후 표준편차로 나눔

$$z = \frac{x - \mu}{\sigma}$$

정규화
데이터의 범위를 0부터 1까지로 변환하여 데이터 분포를 조정
전체 데이터 중에서 해당 값이 어떤 위치에 있는지 파악하는 데에 유용
0에 가까울수록 작은 값, 1에 가까울수록 큰 값
(해당 값 - 최솟값) / (최댓값 – 최솟값)

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

RobustScaler
이상치에 민감한 기본 표준화, 정규화 방식의 단점을 보완한 스케일링 기법
데이터의 중앙값(Q2)을 0으로 잡고 Q1(25%)과 Q3(75%) 사분위수와의 IQR 차이를 1이 되도록 하는 스케일링 기법
이상치의 영향력을 최소화하여 일반적으로 표준화, 정규화보다 성능이 우수

11.5 모델 성능 향상을 위한 파생 변수 생성

파생변수(Derived variable)
원래 있던 변수들을 조합하거나 함수를 적용하여 새로 만들어낸 변수

파생변수는 데이터의 특성을 이용하여 분석 효율을 높이는 것이기 때문에 전체 데이터에 대한 파악이 중요할 뿐만 아니라 해당 비즈니스 도메인에 대한 충분한 이해가 수반되어야 한다.

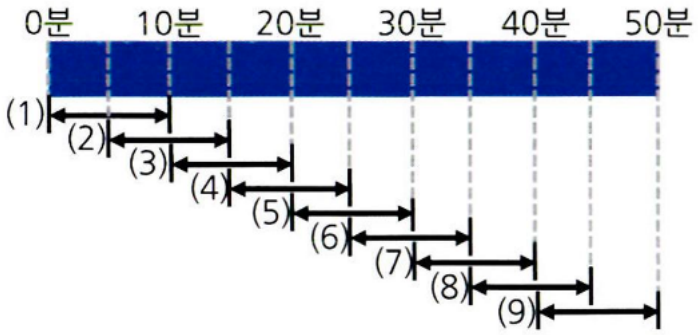
단점)
파생변수는 기존의 변수를 활용해서 만들어낸 변수이기 때문에 다중공선성 문제가 발생할 가능성이 높다.

- ⇨ 파생변수를 만든 다음에는 상관분석을 통해 변수 간의 상관성을 확인
- ⇨ 상관성에 따라 파생변수를 그대로 사용할지, 기존 변수를 제외하고 파생변수만 사용할지 여부를 결정

11.6 슬라이딩 윈도우 데이터 가공

슬라이딩 윈도우(Sliding window)
현재 시점으로부터 ±M 기간의 데이터를 일정 간격의 시간마다 전송하는 방식
각각의 데이터 조각(window)들이 서로 겹치며 데이터가 전송

ex) 총 50분의 시간을 10분 씩 쪼개기 -> 겹치는 구간 없이 깔끔하게 쪼개면 5개의 조각이 되지만, 아래의 그림처럼 5분씩 중첩되도록 조각을 내면 총 9개의 조각이 생성



데이터를 겹치도록 쪼개어 전송하는 이유?
패킷 전송 후, 그 패킷의 전송을 확인 받지 않고도 곧바로 다음 패킷을 보낼 수 있어 네트워크를 효율적으로 사용할 수 있기 때문이다.
But, 데이터를 겹쳐 나눔으로써 전체 데이터가 증가하는 원리를 차용한 것이 슬라이딩 윈도우 데이터 가공의 핵심

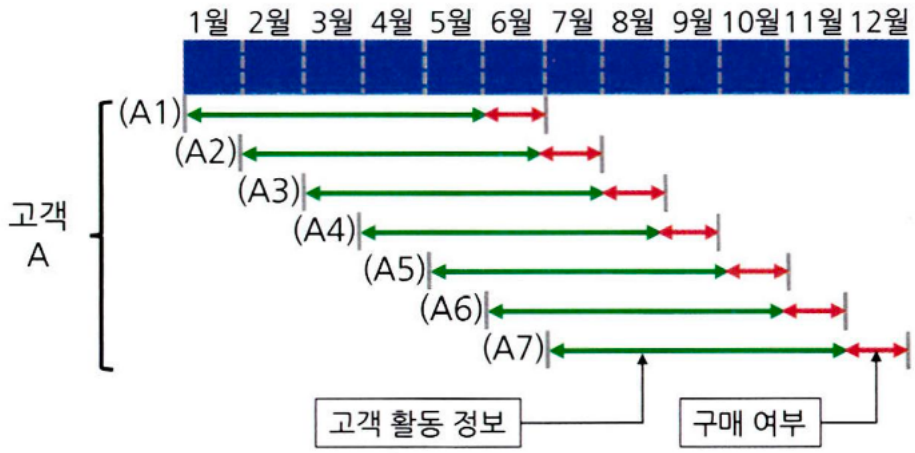
슬라이딩 윈도우 데이터 가공 방법의 쓰임
예측 모델에서 유용

ex) 인터넷 쇼핑몰에서 고객의 지난 5개월 간의 구매내역, 방문 횟수 등의 데이터를 활용하여 한 달 간의 재구매 확률을 구하는 모델을 만든다는 가정.

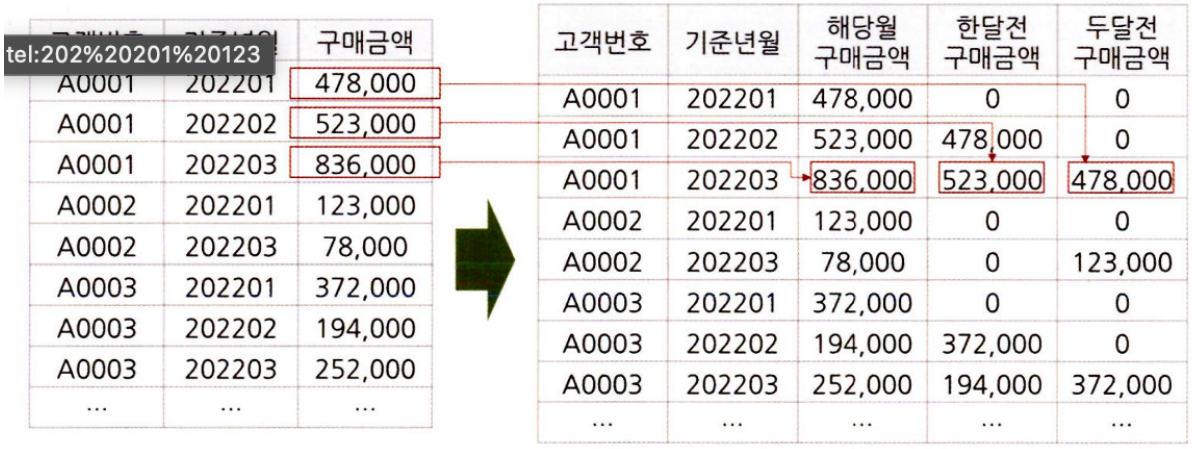
- ⇨ 1년치의 데이터가 있을 경우, 일반적인 방법으로는 6개월을 학습 셋, 나머지 6개월은 테스트 셋으로 데이터를 구성하여 예측 분석 모델을 만든다.

하지만 구매 내역 데이터가 충분하지 않을 경우 예측력이 좋은 모델을 만드는 것이 쉽지 않고, 학습 데이터의 시기와 예측 데이터의 시기와의 시점 차이가 크기 때문에 예측력이 떨어질 위험이 크다.
따라서, 이러한 경우에 슬라이딩 윈도우 방법을 활용하면 많은 분석 데이터셋을 확보하고 학습데이터의 최근성을 가질 수 있다.

아래의 그림처럼 1명의 고객으로 7개의 분석용 데이터를 얻을 수 있는 것이다. 즉, A라는 동일한 사람이 마치 A1, A2, A3...로 시간 차이를 두고 복제되는 것이다. 기존의 1,000건의 고객 정보가 있었다면, 슬라이딩 윈도우 기법을 통해 7,000건의 고객 정보를 확보하게 되는 것이다.



동일한 사람이라도 1월부터 5월 동안의 활동 정보와, 2월부터 6월까지의 정보는 다를 것이다. 물론 성별, 연령 등의 인구 통계학적 정보는 동일하지만, 5개월 동안의 구매내역, 방문 횟수 등은 차이가 있다. 또한 1월부터 5월까지의 활동 후 6월에는 구매를 하지 않았지만, 2월부터 6월가지 활동 후 7월에는 구매를 했을 수 있기 때문에 A1~A7은 서로 다른 관측치로 간주하여 분석 모델에 사용하는 것이 가능하다.



고객들의 월별 구매 금액이 집계된 기존 테이블에서 각 기준년월이 칼럼으로 변환됐다. 즉 매월의 구매금액이 변수가 된 것이다. 주목할 점은 동일한 'A0001' 고객이 2022년 1월 시점에서 해당월, 한 달 전, 두 달 전에 구매한 금액이 생성됐고 2월과 3월 시점의 구매금액들도 생성된 것이다. 이처럼 동일한 고객이 한 달 기간마다 복제되어 데이터셋을 늘릴 수 있다.

11.7 범주형 변수의 가변수 처리

가변수(Dummy variable) 처리
범주형 변수를 0과 1의 값을 가지는 변수로 변환해주는 것.

가변수를 만드는 이유?
범주형 변수는 사용할 수 없고 연속형 변수만 사용가능한 분석기법을 사용하기 위함이다.

➔ 이진변수(binary variable) / 불리언 변수(Boolean variable)

선형 회귀분석이나 로지스틱 회귀분석 등의 회귀분석은 기본적으로 연속형 변수만 사용할 수 있다.

ex) 선형회귀 모델을 통해 고객별 구매금액을 예측하고자 했을 때, 성별을 독립변수로 사용하고자 한다고 가정.
회귀모델은 '남성'과 '여성'을 인식할 수 없다. 차원상의 좌표가 필요하기 때문이다. 따라서 남성은 '0', 여성은 '1'로 변환해 주는 것이다. 혹은 '흡연 여부'를 독립변수로 사용한다고 하면 비흡연은 '0', 흡연은 '1'로 바꿔준다. 일반적으로 해당 안 됨은 '0', 해당됨은 '1'로 처리한다.

성별
여성
남성
남성
여성
여성
남성
…

성별
1
0
0
1
1
0
…

흡연여부
비흡연
흡연
비흡연
비흡연
흡연
비흡연
…

흡연여부
0
1
0
0
1
0
…

그렇다면,
범주가 3개 이상인 경우는 어떻게 가변수 처리를 해줘야 할까? 범주가 늘어날수록, 변수의 수를 늘리면 된다. 예를 들어 혈액형 A, B, AB, O의 범주를 가진 변수를 가변수로 만들고자 한다면, 혈액형 가변수를 3개 만들어 주는 것이다.
여기서 중요한 점은 범주의 개수보다 하나 적게 가변수를 만드는 것이다.

혈액형
B
A
O
AB
A
O
…

혈액형 A	혈액형 B	혈액형 AB
0	1	0
1	0	0
0	0	0
0	0	1
1	0	0
0	0	0
…	…	…

위의 그림에서, 혈액형 범주가 'A형'인 경우는 혈액형 A 가변수의 값이 '1'이고 나머지는 '0'이다. 마찬가지로 혈액형 범주가 'B형'인 경우는 혈액형B 가변수의 값이 '1'이고 나머지는 '0'이다. 그런데 마지막 'O형' 범주는 가변수로 만들어주지 않았다. 왜 그런 것일까? 마지막 범주는 나머지 변수가 모두 '0'이면 해당 범주가 '1'일 것임을 알 수 있기 때문에 굳이 가변수를 만들 필요가 없다.

만약 아무 범주에도 해당 없음도 필요하다면? 그 자체도 하나의 범주이기 때문에 총 범주가 4개가 아닌 5개인 것이고, 가변수도 1개가 적은 4개가 만들어지는 것이다. 어떤 범주를 제거해도 상관 없으나 일반적으로 종속변수에 대한 영향력이 가장 적은 범주를 제거하며, 제거된 범주를 baseline이라 한다. baseline 범주를 기반으로 각각 범주들의 종속변수에 대한 영향력을 산출한다 baseline 범주의 종속변수에 대한 영향력은 0으로 맞춰지며, baseline 범주 대비 다른 범주들의 영향력이 산출되는 것이다.

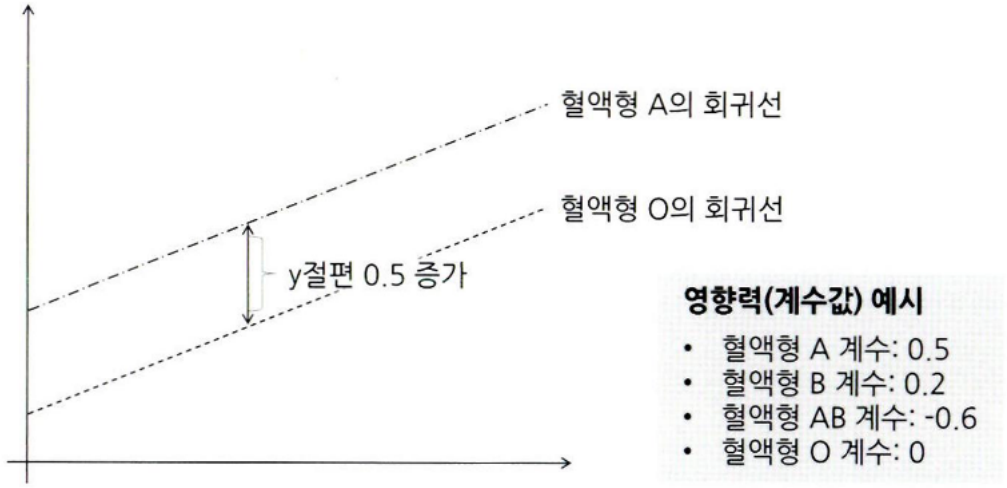
$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + \cdots$$

혈액형 A	혈액형 B	혈액형 AB
1	0	0
0	1	0
0	0	1
0	0	0
…	…	…

Baseline 혈액형 O : 영향력 0

- 혈액형 A는 혈액형 O에 비해 b_1만큼 y에 대해 영향력이 있고,
- 혈액형 B는 혈액형 O에 비해 b_2만큼 y에 대해 영향력이 있고,
- 혈액형 AB는 혈액형 O에 비해 b_3만큼 y에 대해 영향력이 있다.

가변수는 연속형이 아닌 '1'과 '0'으로만 이루어진 변수이기 때문에 회귀선의 기울기는 바꾸지 않고 절편만을 바꾸어 평행하게 움직이도록 만든다. 예를 들어 'A형'의 종속변수에 대한 영향력(계숫값)이 +0.5라면, 회귀선의 절편이 0.5만큼 증가하는 것이다.



위와 같이, 가변수가 범주의 수보다 하나 적게 만들어지는 것은 꼭 데이터의 효율성 때문만은 아니다.

가변수 처리를 하는 것은 기존 하나의 변수를 여러 개의 변수로 나눠준 것이다. 그랬을 때 각각의 변수는 *독립성(independency)*을 가지고 있어야 한다. 쉽게 말해, 독립변수 간에는 서로 영향을 주지 않아야 한다.

만약 독립변수 간에 강한 상관성이 존재하게 되면 다중공선성(Multicollinearity) 문제가 발생한다.

11.8 클래스 불균형 문제 해결을 위한 언더샘플링과 오버샘플링

데이터의 불균형이 심하게 되면 원하는 방향으로 학습이 제대로 이루어지지 않아 예측 정확도가 떨어지게 된다.
왜 데이터 불균형이 심하면 원하는 대로 학습이 이루어지지 않을까?
근본적인 이유는 대부분의 분류 모델에서 적은 비중의 클래스를 분류하는 것이 중요하기 때문이다.

일반적인 기계학습 분류 모델은, 적은 비중의 클래스나 큰 비중의 클래스와 같은 중요도에 차별을 두지 않고 전체적으로 분류를 잘 하도록 학습된다.

데이터 불균형 문제를 해결하는 방법:

1. 모델 자체에 중요도가 높은 클래스에 정확도 가중치를 주어, 특정 클래스의 분류 정확도가 높아지도록 조정해주는 것이다.

⇨ 가중치 밸런싱(Weight balancing)

모델은 손실을 계산하여 손실이 최소화되도록 학습하는데, 가중치 밸런싱은 중요도가 높은 클래스를 잘못 분류하면 더 큰 손실으 계산하도록 조정

2. 불균형 데이터 자체를 균형이 맞도록 가공한 다음 모델을 학습하는 것이다.

⇨ 언더샘플링(Under sampling)

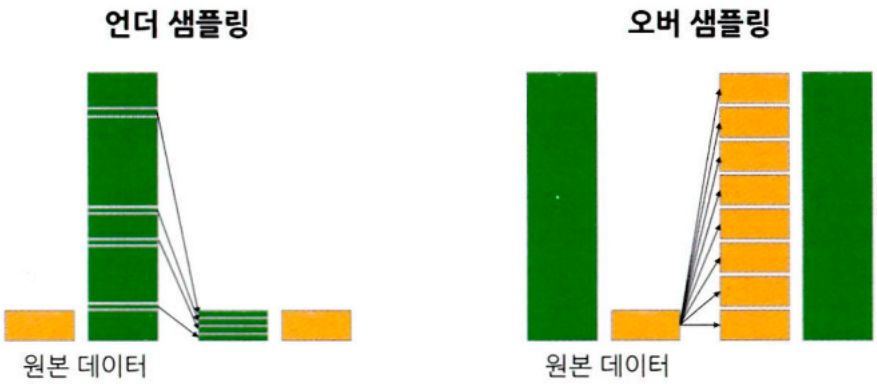
큰 비중의 클래스 데이터를 작은 비중의 클래스 데이터만큼만 추출하여 학습시키는 것

- 랜덤 언더샘플링
- EasyEnsemble
- Condensed Nearest Neighbor(CNN)

⇨ 오버샘플링(Over sampling)

비중이 작은 클래스의 관측치 수와 동일하도록 작은 비중의 클래스의 관측치들을 증가시키는 것.

- 랜덤 오버샘플링
- Synthetic Minority Over-Sampliing Technique(SMOTE)
- Adaptive Synthetic Sampling Approach(ADASYN)



주의사항)

오버샘플링을 적용할 때에는 먼저 학습 셋과 테스트 셋을 분리한 다음 적용(학습 셋과 테스트 셋에 동일한 데이터가 들어가서 과적합을 유발하기 때문)

학습된 모델의 예측력을 검증할 때 사용하는 테스트 셋에는 오버샘플링을 적용하지 않은 순수한 데이터를 사용

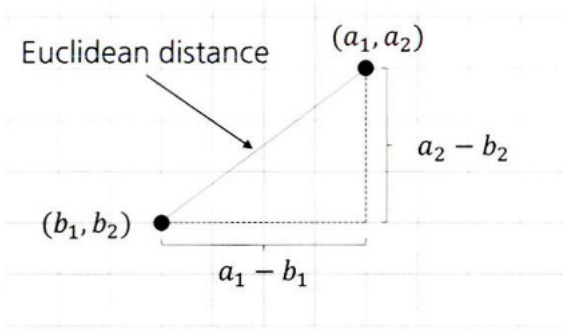
오버샘플링이나 언더샘플링을 적용했을 때는 그렇지 않은 경우보다 예측 성능의 편차가 증가(설정된 알고리즘 seed 값에 따라 데이터의 값이 변하기 때문)

> 오버샘플링이나 언더샘플링을 적용했을 때는 모델 성능 지표를 확인할 때, 여러 번 테스트하여 표준편차와 같은 평가 측도의 변동에 대한 정보를 같이 표기하는 것이 좋다.

11.9 데이터 거리 측정 방법

데이터 간의 '거리(distance)'란?
X, Y 축의 2차원 좌표가 있다고 했을 때, A(1, 1), B(1, 3), C(2, 5)라 하자.
이 때, 관측치 A를 기준으로, B와 C 중 어느 관측치가 더 가까이 있는가를 판단하기 위한 것이 데이터 거리 측정이다(= 데이터 유사도(similarity) 측정).
사용 예)
분류모델, 군집모델, 추천 시스템, etc.

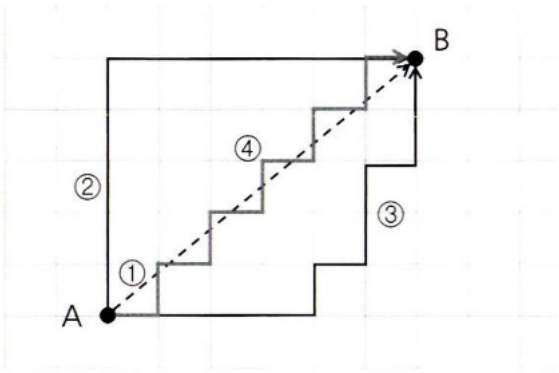
유클리드 거리(Euclidean distance)
피타고라스 정리 활용. 즉, 관측치 간의 직선 거리를 측정.
유클리드 거리 값이 0에 가까울수록 데이터 간의 거리가 짧다는 것이므로, 유사도가 높음을 의미한다.



n차원의 데이터에 대한 유클리드 거리 계산

$$d(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

맨해튼 거리(Manhattan distance)
택시 거리라고도 불리는 맨해튼 거리는 체계적인 도시계획으로 구성된 맨해튼의 격자 모양 도로에서 최단거리를 구하는 원리를 이용한다.
맨해튼 거리는 L1 Norm이라 불리며, L2 Norm은 유클리드 거리다.
보통 딥러닝 분야에서 정규화(Regularization)를 할 때 L1 Norm, L2 Norm이라는 용어로 데이터(벡터) 간 거리를 구한다.



A 지점에서 B 지점까지 최단거리로 가려면?
A 지점과 B 지점까지의 X축 거리, Y축 거리의 합

$$d(A,B) = |a_1 - b_1| + |a_2 - b_2| + \cdots + |a_n - b_n|$$

민코프스키 거리(Minkowski distance)
옵션값을 설정하여 거리 기준을 조정할 수 있는 거리 측정 방법

$$d(A,B) = (\sum_{i=1}^{n} |a_1 - b_1|^p)^{\frac{1}{p}}$$

수식에서처럼, 유클리드 거리 수식과 동일하며 단지 제곱 부분을 p-norm값으로 하여 조정 가능
p값을 1로 설정하면 맨해튼 거리와 동일하고, 2로 설정하면 유클리드 거리와 동일하다. P값은 반드시 1 이상이어야 하고 정수가 아니어도 상관이 없다.

체비쇼프 거리(Chebyshev distance)
맥시멈 거리(Maximum distance), 혹은 L max Norm으로도 불린다.
군집 간 최대 거리를 구할 때 사용.
군집 간의 최대 거리가 중요한 경우에 사용하며 이 역시 계산 값이 0에 가까울수록 유사한 것이다.

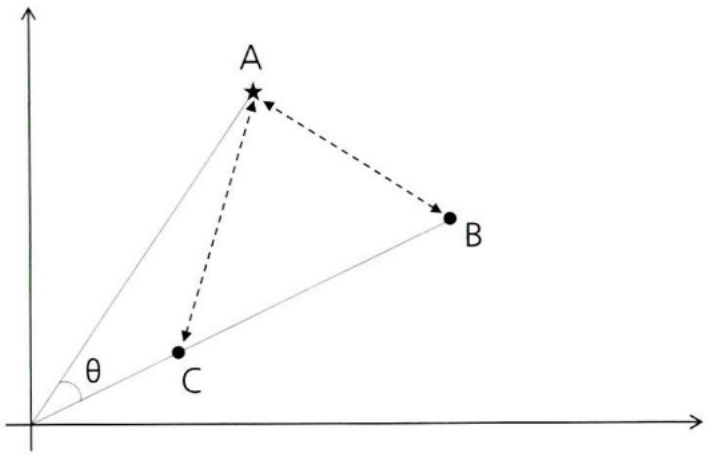
$$d(A,B) = \max(|a_1 - b_1|) = \lim_{n\to\infty}[\sum_{i=1}^{p}|a_i - b_i|^n]^{\frac{1}{n}}$$

마할라노비스 거리(Mahalanobis distance)
유클리드 거리에 공분산을 고려한 거리 측정 방법
변수 내 분산과 변수 간 공분산을 모두 반영하여 A와 B 간 거리를 계산
단순 거리에 상관성을 함께 볼 수 있다는 장점
확률 분포를 고려하기 때문에 공분산 행렬을 더해준다.

$$d(A,B) = \sqrt{(A-B)^{\Sigma^{-1}}(A-B)^T}$$

Σ^{-1} : 행렬의 역행렬
T : 변환행렬

코사인 거리(Cosine distance)
코사인 유사도(Cosine Similarity)는 두 벡터의 사이각을 구해서 유사도를 구하는 것이다.
아래의 그림처럼 두 점 간의 각도가 작으면 유사도가 높고, 각도가 크면 유사도가 낮아진다.



코사인 유사도는 -1에서 1 사이의 값을 가지며, 두 벡터의 방향이 완전히 동일하면 1의 값을 가진다. 반대로 180 각도는 코사인 유사도가 -1이 되며, 90은 0이 된다. 1은 서로 유사도가 매우 높

음, 0은 관련성이 없음, -1은 완전히 반대됨을 의미한다.
하지만 코사인 유사도를 사용하는 경우는 양수만 갖는 변수 혹은 0과 1의 값으로만 이루어진 변수를 사용하기 때문에 일반적으로 코사인 유사도는 0~1의 값을 갖는다.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

두 변수의 벡터 내적을 구한 뒤, 두 벡터의 각각의 크기를 구하여 곱한 값으로 나눈다. 분자 부분은 벡터 내적으로, 각 변수들을 순서대로 곱한 다음 그 결과들을 모두 더한다. 분모 부분은 벡터의 크기(norm)를 곱한 것이로, 피타고라스 정리에서 직각삼각형의 빗변을 구하는 것과 같다.