



Chapter 05. 분류 - 설유환

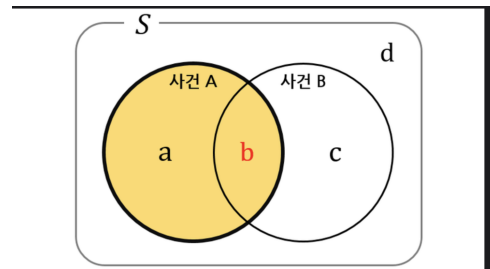
- Multivariate \Rightarrow 0, 1, 2 예측이라면 0인지 0보다 큰지 먼저 하고, 1인지 2인지 예측

5.1 나이브 베이즈 . Naive Bayes

조건부확률

조건부 확률은 사건 B가 일어나는 경우에 사건 A가 일어날 확률을 말한다. 사건 B가 일어나는 경우에 사건 A가 일어날 확률은 $P(A | B)$ 로 표기한다. 사건 B가 발생했을 때 사건 A가 발생할 확률은 사건 B의 영향을 받아 변하게 된다.

즉, 한 사건이 일어났다는 전제 하에서 다른 사건이 일어날 확률



$$P(B|A) = \frac{n(A \cap B)}{n(A)} = \frac{b}{a+b}$$

위 벤다이어그램에서 $N = a + b + c + d$ 라 하면,

$$P(B|A) = \frac{b}{a+b} = \frac{\frac{b}{N}}{\frac{a+b}{N}} = \frac{P(A \cap B)}{P(A)}$$

전체 중고차 중 70%가 에어컨이 있고 40%가 CD 플레이어가 있다고 하자. 전체 중고차 중 90%가 둘 중 적어도 하나는 가지고 있다고 한다. 중고차 중 임의로 뽑은 한 대가 에어컨이 없는 중고차일 때, 이 차가 CD 플레이어도 없을 확률은?

B를 에어컨이 없는 중고차를 고르는 사건, A를 CD 플레이어가 없는 중고차를 고르는 사건이라고 하면 다음이 성립한다.

$$P(B) = 0.3$$

$$P(A) = 0.6$$

$$P(A \cap B) = 0.1 \text{ (에어컨과 CD 플레이어가 모두 없을 확률)}$$

$$P(A|B) = P(A \cap B)/P(B) = 0.1/0.3 = 1/3$$

최대우도법 Maximum Likelihood Estimation, 이하 MLE

- 우리가 데이터를 관찰함으로써 이 데이터가 추출되었을 것으로 생각되는 분포의 특성을 추정할 수 있음
- likelihood 가능성 \Rightarrow 가능도 : 지금 얻은 데이터가 이 분포로부터 나왔을 가능도
- 수치적으로 이 가능도를 계산하기 위해서는 각 데이터 샘플에서 후보 분포에 대한 높이(즉, likelihood 기여도)를 계산해서 다 곱한 것을 이용할 수 있을 것
- 주황색 likelihood 다 곱한 값이 파랑색 보다 높을 것 \Rightarrow 예측가능
- 오른쪽은 likelihood function(우도 함수, 가능도 함수) 식인데 이걸 최대화 하는게 목적일때 쓰는 방법이 MLE
- 그대로 쓰는건 아니고 Log 취해줌 \Rightarrow 그냥 계산이 더 편해지니까
- 최댓값을 찾는법 \Rightarrow 미분이나 편미분을 이용

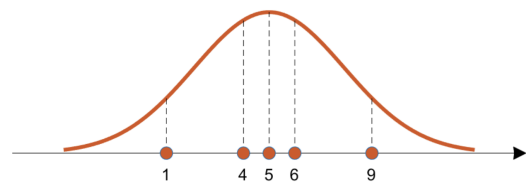


그림 2. 주황색 후보 분포에 대해 각 데이터들의 likelihood 기여도를 점선의 높이로 나타냈다.

$$P(x|\theta) = \prod_{k=1}^n p(x_k|\theta)$$

Maximum Likelihood Estimation

- 어떤 함수의 최댓값을 찾는 방법 \rightarrow 미분(혹은 편미분)을 이용!
- 즉, 찾고자 하는 파라미터 θ 에 대해 다음과 같이 편미분하여 최댓값을 찾을 수 있다.

$$\frac{\partial}{\partial \theta} L(\theta|x) = \frac{\partial}{\partial \theta} \log P(x|\theta) = \sum_{i=1}^n \frac{\partial}{\partial \theta} \log P(x_i|\theta) = 0$$



용어정리

📌 조건부확률 (conditional probability) : 어떤 사건이 주어지면 해당사건을 관찰할 확률

📌 사후확률 : 예측 정보를 통합한 후 결과의 확률(이와 달리, 사전확률에서는 예측 변수에 대한 정보를 고려하지 않는다).

베이즈 정리

베이즈 정리의 공식

- 베이즈 정리의 공식은 아래와 같다.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- 이 공식에는 네 개의 확률이 포함되어 있음.
- 이 중 $P(H)$ 와 $P(H|E)$ 는 각각 사전확률, 사후확률이라고 불림.
- 기본적으로 베이즈 정리는 사전확률과 사후확률의 관계에 대해 설명하고 있음.

-
- 기본적으로 사전확률과 사후확률의 관계에 대해 설명하고 있음
-

- 전통적으로 빈도주의 \Rightarrow 동전뒤집기 10,000번 던지면 5,000번은 앞면이 나온다. (거의 대부분 확률론과 통계학을 지배하는 관점)
- 새로운 베이지안 주의 \Rightarrow 동전의 앞면이 나왔다는 이 주장에 대한 신뢰도가 50%이다. (즉 확률을 주장에 대한 신뢰도로 해석하는 관점)

용어 정리

- 아래의 공식에서 E와 H가 구체적으로 의미하는 것은?

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- H: Hypothesis. 가설 혹은 '어떤 사건이 발생했다는 주장'
- E: Evidence. '새로운 정보'
- P(H): 어떤 사건이 발생했다는 주장에 관한 신뢰도
- P(H|E): 새로운 정보를 받은 후 갱신된 신뢰도

- 가설 혹은 주장은 어떠한 가설이든 사건이든 상관없음

용어 정리

- P(H): 어떤 사건이 발생했다는 주장에 관한 신뢰도
- P(H|E): 새로운 정보를 받은 후 갱신된 신뢰도

$$\underset{\text{사후 확률 (posterior)}}{P(H|E)} = \frac{P(E|H) \overset{\text{사전 확률 (prior)}}{P(H)}}{P(E)}$$

베이즈 정리는 사전확률과 사후확률 간의 관계에 대해 설명하는 정리이다

- 사전확률을 가지고 있는데, 이걸 불확실성이 있다.
- 이때 사건이나 근거를 가지고 신뢰도를 어떻게 갱신할수 있는지 그것에 대한 수학적인 알고리즘을 밝혀 놓은것이 베이즈정리
- 연역적 추론 ⇒ 귀납적 추론으로

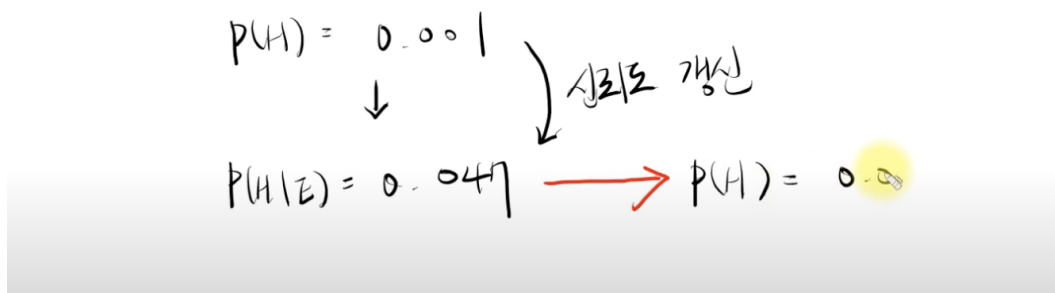
(2종 오류는 비용을 많이 쓰게될 확률 = 드래곤이 나타나 우리를 다 죽일 것이라는 가설 - 미스릴 갑옷을 구매해야 한다.)

우리는 $P(E)$ 를 모르는데 $P(E)$ 는 \Rightarrow 파란색 박스와 초록색 박스의 합
(공식이 나오는데 공식은 나도 너무 어려워서 이해하지 못했...음)

- 즉 파란 + 초록 합 분의 파란색 넓이가 베이즈 정리

예제 2

- 예제 1에서 한 번 양성 판정을 받았던 사람이 두 번째 검진을 받고 또 양성 판정을 받았을 때, 이 사람이 실제로 질병에 걸린 확률은?


$$\begin{array}{ccc} P(H) = 0.001 & \downarrow & \\ & & \text{신뢰도 갱신} \\ P(H|E) = 0.047 & \longrightarrow & P(H) = 0.047 \end{array}$$

5.1.1 나이브 하지 않은 베이즈 분류는 왜 현실성이 없을까 ?

나이브 naive

1. [못마땅함](경험·지식 부족 등으로) 순진해 빠진, (모자랄 정도로) 순진한

to be politically naive : 정치적으로 순진해 빠졌다[모자라다]

- 예측변수의 개수가 일정 정도 커지게 되면 분류해야 하는 데이터들은 대부분 서로 완전 일치하는 경우가 거의 없다.

1. 나이브 베이즈는 클래스 자체가 아닌 클래스들이 갖는 확률을 리턴한다.
2. 베이즈 정리는 어떤 이벤트와 관련된 조건에 대한 사전 믿음으로 그 이벤트가 발생할 확률을 표현한다.

a. 양성 판정을 받았을 때 진짜 병 걸렸을 확률을 베이즈 정리로 알아보자.

- 컨디션이 별로라서 병원에 갔더니 천명 중 한명 걸린다는 희귀병 xx병 테스트 결과 양성 땀.
- 이거 얼마나 정확한 거임? 의사 왈: 병 걸린 사람이 테스트하면 정확하게 분류할 확률이 99%임
- 그러나! 내가 그 병에 걸렸을 확률이 99%를 의미하는 것이 아님.
- 왜? “병 걸린 사람이” 라는 조건이 붙었기 때문. 나는 병에 걸렸는지 모르니까 조건이 안 걸린 상태임.

$$P(\text{멀쩡}) = 1 - P(\text{병}), P(\text{양성}|\text{멀쩡}) = 1 - p(\text{양성}|\text{병})$$

- 그래서 값을 다 집어넣으면 양성 결과가 나왔을 때 실제로 병에 걸렸을 확률은 9%가 된다.
- 여기서 병에 걸릴 확률을 사전 확률이라 함. 병에 걸릴 확률을 어떻게 알겠음. 신도 아니고. 그래서 전체 인구 중에 걸린 사람 수를 나눠서 적절히 구한 것임.
- 전체 인구 수를 기준으로 병에 걸릴 사전 확률은 0.1%에 불과했으나, 한번 양성인 따니까 9%로 병에 걸릴 확률이 올라갔음. 이를 사후 확률이라고 함.
- 베이즈 정리는 이와 같이 사전 믿음(병 걸릴 확률)을 새로운 정보(테스트 결과)를 사용해 새로운 사후 확률(테스트 결과를 보고 났더니 병에 걸릴 확률)로 업데이트하는 것임
- 이 의사 못 믿겠어서 다른 병원에서 테스트를 한번 더 봤는데 다시 양성인 떴다. 이럴 때 병에 걸렸을 확률은?
- 여기서 사전 확률은 이전 테스트를 통해 얻은 사후 확률이 된다. 병에 걸렸을 때 테스트가 양성일 확률이 99%로 동일하다고 하면 $P(\text{병}|\text{양성})$ 은

$$P(\text{병}|\text{양성}) = \frac{0.99*0.09}{0.09*0.99+0.91*0.01} \approx .91$$

- 새로운 사전 확률로 업데이트한 사후 확률은 91%. 미심쩍으면 두번 테스트해보면 되겠다.

3. 나이브 베이즈는, 분류 모델로 설명하자면, 인풋이 주어졌을 때 타깃 클래스들의 확률을 출력한다.

4. 나이브 ?

재는 참 사람이 나이브해 하면 만사를 조금 너무 쉽게 보고 대충대충 한다는 그런 느낌이 있다.

나이브 베이즈는 뭘 대충 하길래 나이브라는 이름이 붙었을까??

- 바로 직전에 베이즈 정리를 이용해서 메일에 등장하는 단어들로 스팸을 예측하는 수식을 유도했다.

$$P(\text{단어1}|\text{단어2}\text{단어3}\text{스팸}) \cdot P(\text{단어2}|\text{단어3}\text{스팸}) \cdot P(\text{단어3}|\text{스팸}) \cdot P(\text{스팸})$$

체인 룰로 결합법칙을 풀어버리면서 단어들간에 의존관계가 생겼다.

“다이어트에 딱 좋은 이 약을 구매하세요”와 “나 어제부터 다이어트하는데 개실패함”이라는 두 문장이 있을 때 둘 다 **다이어트**가 사용되었으나, 두번째는 스팸 냄새가 덜 난다. 근처의 단어들이 **다이어트**가 갖는 속성에 영향을 주기 때문.

따라서 위에 풀어쓴 수식처럼 단어간의 의존 관계를 반영한 모델을 만드는 것이 필요해 보인다.

그리고 나이브 베이즈는 그 의존관계를 깡그리 무시하고, 단어들은 서로 완전히 독립적이라는 다소 순수한, 즉 나이브한 가정을 베이즈 정리에 적용하기 때문에, 나이브 베이즈라는 이름이 붙은 것이다.

피쳐간의 관계를 독립적이라고 가정해버리는 나이브 베이즈의 선택은 일견 데이터의 특성을 온전히 반영하지 못하는 듯 하다. 그러나 실제로 스팸분류기 등의 분류 모델에서 나이브 베이즈가 어느정도 성능이 잘 나오는 것을 보면,

- (1)실제로 피쳐간 관계를 따질 만큼 문제가 복잡하지 않거나 (특정 단어의 등장만으로도 판단할 수 있거나) 혹은
- (2) 독립성 가정으로 파라미터가 적은 모델이 상대적으로 노이즈에 더 강하기 때문이지 않을까.

참고

- 나이브 베이즈는 베이즈 통계의 방법으로 분류되지 않는다. ⇒ 통계가 거의 필요없고 경험적
 - 베이즈 규칙과 비슷한 예측이 들어가다 보니 이름을 그렇게 붙였을뿐

5.1.2 나이브한 해법

- 나이브 베이즈 방법에서는 확률을 계산하기 위해 정확히 일치하는 레코드로만 제한할 필요가 없다

5.1.3 수치형 예측변수

- 베이즈 분류(기)는 예측변수들이 카테고리컬인 경우 적합함
 - (스팸 분류에서 특정 단어, 어구, 문자열의 존재여부등)

사이킷런에서 제공하는 나이브베이즈 모형

사이킷런의 `naive_bayes` 서브패키지에서는 다음과 같은 세가지 나이브베이즈 모형 클래스를 제공한다.

- `GaussianNB`: 정규분포 나이브베이즈
- `BernoulliNB`: 베르누이분포 나이브베이즈
- `MultinomialNB`: 다항분포 나이브베이즈

이 클래스들은 다양한 속성값 및 메서드를 가진다. 우선 사전 확률과 관련된 속성은 다음과 같다.

- `classes_`
- 종속변수 Y의 클래스(라벨)
- `class_count_`
- 종속변수 Y의 값이 특정한 클래스인 표본 데이터의 수
- `class_prior_`
- 종속변수 Y의 무조건부 확률분포 $P(Y)$ (정규분포의 경우에만)
- `class_log_prior_`

- 종속변수 Y 의 무조건부 확률분포의 로그 $\log P(Y)$ (베르누이분포나 다항분포의 경우에만)

5.2 판별분석



용어정리

- 📌 공분산 : 하나의 변수가 다른 변수와 함께 변화하는(유사한 크기와 방향) 정도를 측정하는 지표
- 📌 판별함수 : 예측변수에 적용했을때 클래스 구분을 최대화 하는 함수
- 📌 판별 가중치 : 판별함수를 적용하여 얻은 점수를 말하며, 어떤 클래스에 속할 확률을 추정하는데 사용된다.

5.2.1 공분산 행렬

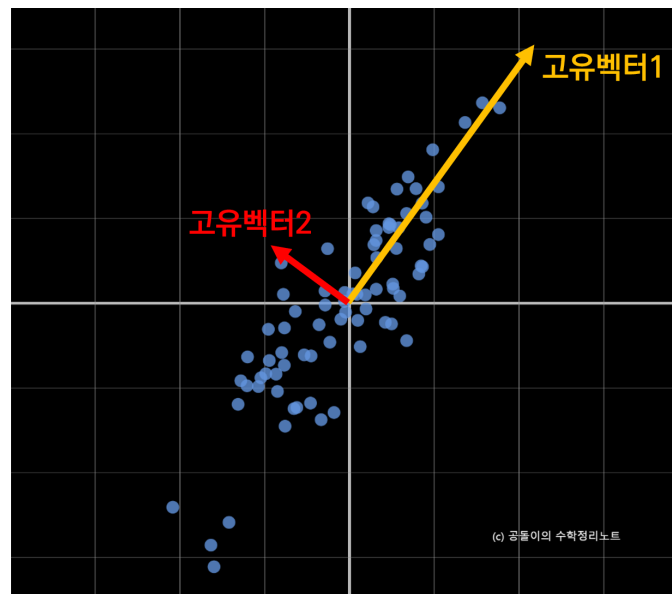
- 공분산 covariance 이란 두 변수 X 와 z 사이의 관계를 의미하는 지표
- 공분산 행렬은 일종의 행렬로써, 데이터의 구조를 설명해주며, 특히 특징 쌍(feature pairs)들의 변동이 얼마나 닮았는가(다른 말로는 얼마만큼이나 함께 변하는가)를 행렬에 나타내고 있다.
- . 행렬이란 선형 변환이고 하나의 벡터 공간을 선형적으로 다른 벡터 공간으로 mapping 하는 기능을 가진다.



그림 4. 공분산행렬 Matrix 1의 각 원소들이 의미하는 것

- 고유 벡터(eigenvector)의 의미를 잘 생각해보면, 고유 벡터는 그 행렬이 벡터에 작용하는 주축(principal axis)의 방향을 나타내므로 공분산 행렬의 고유 벡터는 데이터가 어떤

방향으로 분산되어 있는지를 나타내준다고 할 수 있다.



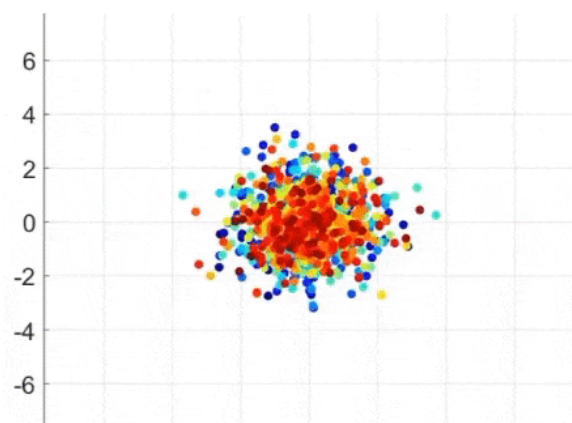
Covariance Matrix는 두 가지 개념으로 바라볼 수 있는데

첫번째는 구조적인 의미로 각 feature들의 퍼져있음이 얼마나 유사하냐 (feature의 변동이 얼마나 닮았냐) 이다. 즉, 각 feature간의 내적을 통해 feature의 변동이 얼마나 닮았는지를 알 수 있다.
(키 vs 몸무게, 몸무게 vs 성적,)



두번째는 수학적 (기하학적) 의미로 데이터를 어떻게 linear transform하고 있는가이다.

행렬은 linear transform을 통해 다른 벡터 공간으로 mapping해주는 기능을 가지는데, 이 covariance matrix는 (각자의 데이터가 서로 관련이 없는) 초기 상태에서 서로의 연관성에 대한 정보가 담겨져 있는 covariance matrix를 통해 각 데이터를 분산시켜 준다고 볼 수 있다. 이 분산시키는 형태는 마치 특정 방향으로 잡아 늘리는 (shearing) 모습을 하고 있다.



(첫번째 구조적인 의미는 이미 퍼져있는 상태에서의 관계를 봤고여기서는 데이터가 어떻게 “퍼져 나가는지”를 바라봤음)

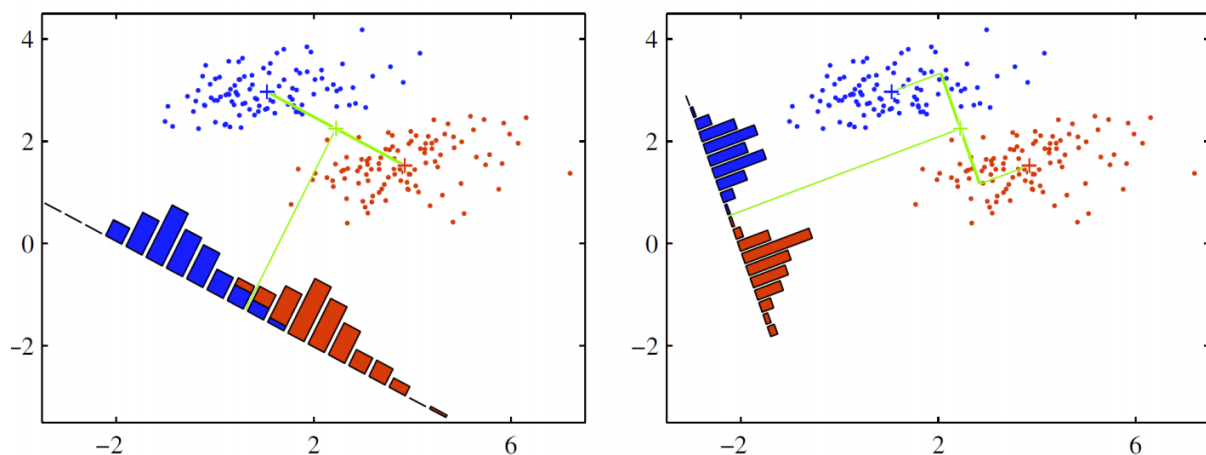
5.2.2 피셔의 선형판별

LDA (linear discriminant analysis) - 피셔의 해법

Fisher Discriminant Analysis란?

FDA 혹은, Linear Discriminant Analysis(**LDA**)라고 불린다.

데이터들을 하나의 직선(1차원 공간)에 projection시킨 후 그 projection된 data들이 잘 구분이 되는가를 판단하는 방법이다.



데이터가 잘 구분되어있다는 의미는 위 그림 중, 왼쪽 보단 오른쪽 처럼 구분이 되어야 됨을 의미한다. 이 특성을 보자면, 데이터들이 모여있고, 중심부가 서로 멀수록 데이터의 구분이 잘 됐음을 알 수 있다. 즉, projection 후 두 데이터들의 중심(평균)이 서로 멀수록, 그 분산이 작을수록 구분이 잘 되었다고 얘기할 수 있다. 이렇게 잘 분류되게끔 하는 하나의 vector w 를 구하는 것이 LDA이다.

5.3 로지스틱 회귀



용어정리

- 📌 로짓 : 0~1이 아니라 +- 무한대의 범위에서 어떤 클래스에 속할 확률을 결정하는 함수
- 📌 오즈 : 실패에 대한 성공의 비율
- 📌 로그 오즈 : 변환 모델(선형)의 응답변수, 이 값을 통해 확률을 구한다.

- 데이터 위주의 접근방식이라기 보다 구조화된 모델 접근방식

5.3.1 로지스틱 반응함수와 로짓

- 오즈비 odds rate \Rightarrow 성공과 실패의 비율, 사건이 발생할 확률과 하지 않을 확률
 - 이길확률과 이기지 못할확률 같은 예시가 있다.

$$g(x) = \frac{e^x}{1 + e^x}$$

시그모이드(sigmoid) 함수

하루에 담배 10개피를 피운 사람 a를 조사했더니 폐암에 걸렸다(1). 그런데 하루에 담배 10개피를 피운 사람을 10명 조사했더니 6명은 폐암에 걸렸으나 4명은 걸리지 않았다. 결국 10개피의 담배를 피운 사람은 60%의 확률로 폐암에 걸렸다고 말할 수 있는 것이다.

이처럼 조사 횟수가 많아지게 되면 종속 변수 y는 확률로 표현할 수 있게 되는 것이며 이럴 경우에도 종속 변수의 값은 0 ~ 1의 범위(확률상 0% ~ 100%)를 넘지 않는다.

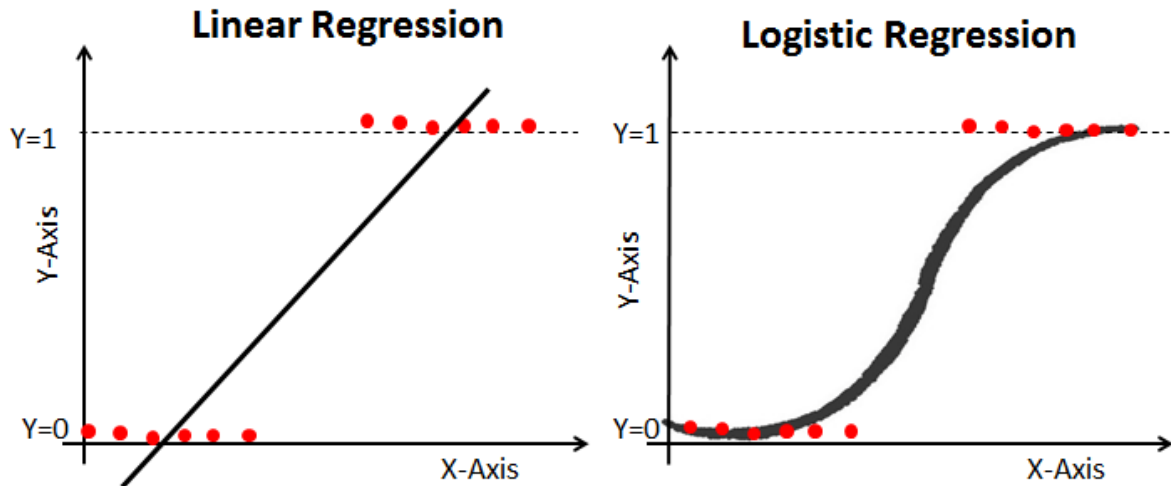
로지스틱회귀분석은 종속변수가 범주형이면서 0 or 1 인 경우 사용하는 회귀분석이다.

로지스틱 회귀분석을 설명하기 위해서는 먼저

로지 변환

과

에 대해서 알아야한다.



왼쪽그림의 경우, Y가 0 or 1(사망/생존, 실패/성공, 불합격/합격)이라면

선형회귀로는 fitting하기가 힘들다

. 그래서

곡선으로 fitting 하기 위해 사용하는 것이 **logistic함수 = 로짓변환** 이다.

Odds 는 비 이고, Odds ratio 는 비의 비율 이다. Odds의 해석은 확률 에서 시작되어 실패에 비해 성공할 확률의 비 를 의미하며 , $odds = p / 1-p$ 로 계산한다. 예를 들어, 게임에서 이길 확률이 1/5, 질 확률이 4/5이면, 게임에서 이길 odds 1/4이 되며, 계산된 값을 바탕으로 5번 중에, 4번 질 동안 1번 이긴다 라고 해석한다.

이 Odds에 Log를 취한 것 이 바로 로짓(Log $p/1-p$) 이다.

예를 들어, 아래와 같은 교차표가 있다고 가정해보자. Odds라는 것은 각 독립변수(drugA, drugB)에 대해 실패/성공에 대한 확률을 구한 뒤, 각각 구하는 것이며,

<생존Odds>가 구해지면 -> Odds ratio도 계산 할 수 있다.

Odds ratio는 위험요인과 질병발생간의 연관성을 나타낼 때 사용 + 논문기재시 신뢰구간도 같이 제시 해야한다. 예를 들어, 대조군(DrugB)와 실험군(Drug A)를 이용해서, 위험요인(Drug A)와 생존/사망의 연관성을 나타낼때, 각각의 Odds -> Odds ratio를 구한 뒤 제시한다.

오즈비 = 교차비 = 승산비 = 대응위험도 라는 표현도 쓴다.

5.3.2 로지스틱 회귀와 GLM

일반선형회귀의 경우 선형성, 독립성, 등분산성, 정규성의 가정을 갖고 있습니다. 하지만, 종속변수가 연속형이 아니라면 대표적으로 오차항의 정규성(오차항의 분포는 정규성(Normality)을 가져야 한다.) 가정이 깨지게 됩니다.



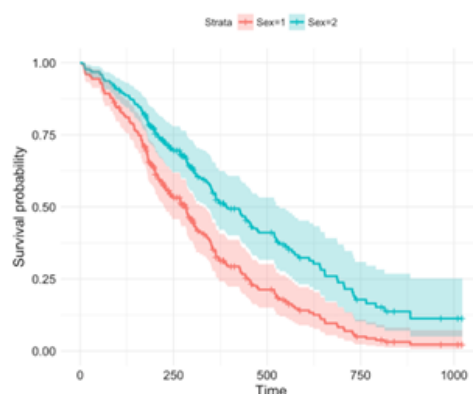
오차항의 정규성

이 가정은 모든 xx 값에 대해서 데이터가 회귀선을 기준으로 정규분포를 따르게 분포되어야 한다는 것이다. 즉, 회귀선을 기준으로 그 주위에 더 많은 데이터가 있고, 이상치가 없이 균일하게 퍼져있어야 한다.

이런 경우, 종속변수에 맞게 대표적으로 로지스틱회귀(Logistic Regression)과 Cox의 비례위험회귀(Cox's Proportional Hazard Regression)이라는 대표적인 일반화선형회귀를 사용합니다.

일반화선형회귀는 종속변수를 적절한 함수로 변화시킨 $f(y)$ 를 독립변수와 회귀계수의 선형 결합으로 모형화한 것

Cox의 비례위험회귀(Cox's Proportional Hazard Regression)



Cox의 비례위험회귀는 시간에 따라 hazard ratio가 일정하다는 가정을 갖은 생존분석 중 가장 많이 쓰이는 방법론으로서, 어떤 사건(event)이 일어날 때까지의 시간을 대상으로 분석하는 통계방법입니다.

일반화선형회귀 : 종속변수를 변환하여, 회귀계수를 독립변수의 선형결합으로 나타낸 모형

참고

1. Odds(A)구하기

- $P(A, A\text{에 대해 생존확률}) = 20 / 52 = 0.38$
- $1 - P(A, 사망확률) = 0.62$
- $\text{Odds}(A) = \text{실패(사망)에 비해 생존(성공)할 확률의 비} = 0.38 / 0.62 = 0.61$
- A약 먹으면, 100명 사망할 동안, 61명 생존

2. Odds(B)구하기

- $P(B, 생존/성공확률) = 24/66 = 0.63$
- $1 - P(B, 사망/실패확률) = 0.37$
- $\text{Odds}(B) = 0.63 / 0.37 = 1.7$
- B약 먹으면, 100명 사망할 동안, 170명 생존

3. Odds ratio 구한 뒤 해석 하기

- $B\text{에 대한 } A\text{의 Odds ratio} = 0.61 / 1.7 = 0.36$
- 해석 : B에 비해 A일 때, 생존(성공)이 0.36배 = 64%가 생존율(성공률)이 떨어진다는 점

5.3.5 계수와 오즈비 해석하기

- 재계산 없이 새로운 데이터에 대해 결과를 빨리 계산할 수 있다는 점

5.4 분류 모델 평가하기



용어정리

- 📌 정확도 :
- 📌 혼동행렬 : 분류에서 예측된 결과와 실제 결과에 대한 레코드의 개수를 표시한 테이블
- 📌 민감도 특이도 정밀도
- 📌 ROC 곡선 : 민감도와 특이성을 표시한 그림
- 📌 리프트 : 모델이 다른 확률 컷오프에 대해 1을 얼마나 더 효과적으로 구분하는지 나타내는 측정지표

- $\text{정확도} = \frac{\text{참양성} + \text{참 음성}}{\text{표본크기}}$

5.4.1 혼동행렬

혼동행렬 (Confusion Matrix)

		Condition (실제)	
		Positive	Negative
Prediction (예측)	Positive	TP - True Positive (참 양성)	FP - False Positive (긍정 오류)
	Negative	FN - False Negative (부정 오류)	TN - True Negative (참 음성)

5.4.2 희귀 클래스 문제

- 클래스간에 불균형 존재 \Rightarrow 쉽게 분류하기 어려움 (사기성 보험청구)
- 모델의 정확도만 올리는 것이 능사는 아니고 참 양성을 잘 골라야함

5.4.2 정밀도 재현율 특이도

정밀도

정확도 높음
정밀도 높음



정확도 낮음
정밀도 높음



정확도 높음
정밀도 낮음



정확도 낮음
정밀도 낮음



재현율

정의 [편집]

		실제 정답	
		Positive	Negative
실험 결과	Positive	True Positive	False Positive (Type II error)
	Negative	False Negative (Type I error)	True Negative

통계적 분류 분야에서 정밀도(precision)와 재현율(recall)은 다음과 같이 정의된다.[1]

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

특이도 민감도

		Condition (실제)	
		Positive	Negative
Prediction (예측)	Positive	TP = 20 True Positive	FP = 90 False Positive
	Negative	FN = 10 False Negative	TN = 880 True Negative
		Sensitivity (민감도)	Specificity (특이도)

$$\text{Sensitivity (민감도)} = \text{TP} / (\text{TP} + \text{FN})$$

병에 걸린 사람이 검사 결과 양성으로 나올 확률

$$\text{Specificity (특이도)} = \text{TN} / (\text{TN} + \text{FP})$$

건강한 사람이 검사 결과 음성으로 나올 확률

5.4.4 roc 곡선 + AUC

ROC(Receiver Operating Characteristic) Curve의 구성 요소

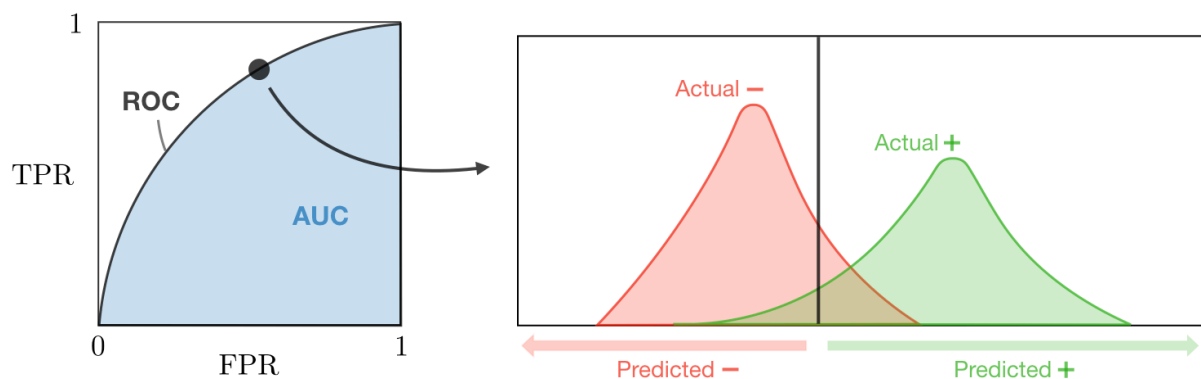
- X축: 1-특이도(False Positive Rate)
- Y축: 민감도(True Positive Rate)
- 특이도: Negative를 Negative로 예측
- 민감도: Positive를 Positive로 예측

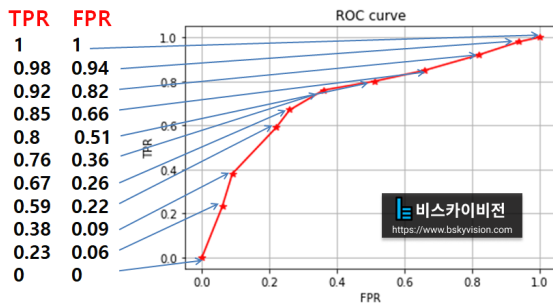
민감도(sensitivity)

- 1인 케이스에 대해 1이라고 예측한 것

특이도(specificity)

- 0인 케이스에 대해 0이라고 예측한 것





$$\text{진양성률(TPR)} = \frac{\text{진양성}}{\text{진양성} + \text{위음성}}$$

$$\text{위양성률(FPR)} = \frac{\text{위양성}}{\text{진음성} + \text{위양성}}$$

Copyright © Gilbut, Inc. All rights reserved.

AUC(Area Under the ROC Curve)

- 진단의 정확도를 측정할 때 사용하는 것으로 ROC Curve아래 면적

AUC 수치 [Ⓢ]	정확성 정도 [Ⓢ]
AUC = 0.5 [Ⓢ]	비 정보적임 [Ⓢ]
0.5 < AUC ≤ 0.7 [Ⓢ]	덜 정확함 [Ⓢ]
0.7 < AUC ≤ 0.9 [Ⓢ]	중등도의 정확성 [Ⓢ]
0.9 < AUC < 1 [Ⓢ]	매우 정확함 [Ⓢ]
AUC = 1 [Ⓢ]	완벽한 검사 [Ⓢ]

5.5 불균형 데이터 다루기

5.5.1 과소표본추출 다운샘플링

1. 데이터가 충분하다면
2. 치우쳐진 0과 1의 균형을 맞출수 있다.

⇒ 정보량이 손실됨 유용한 정보까지 버릴수도

5.5.2 과잉표본추출 업샘플링 . 상향 하향 가중치

⇒ 복원 추출 방식으로 업샘플링 진행

혹은 가중치를 조절한다 .

sample_weight

```
from sklearn import datasets
iris = datasets.load_iris() # iris 데이터를 읽어온다
```

```

X = iris.data
Y = iris.target

enc = OneHotEncoder()
Y_1hot = enc.fit_transform(Y.reshape(-1, 1)).toarray() # 원핫인코딩한다

sample_weight = Y.copy()
sample_weight[sample_weight == 0] = 1 # 0을 1로 바꾼다

# 동일한 가중치로 피팅한다
# model.fit(X, Y_1hot, epochs=300, batch_size=10)

# 샘플별 가중치를 부여하여 피팅한다
model.fit(X, Y_1hot, epochs=300, batch_size=10, sample_weight = sample_weight)

# 클래스별 가중치를 부여하여 피팅한다
model.fit(X, Y_1hot, epochs=300, batch_size=10, class_weight = {0:1, 1:1, 2:2})

```

sample_weight값을 출력해보면 아래와 같습니다.

```

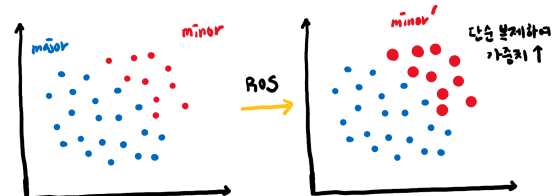
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

```

5.5.3 데이터 생성

Random Over Sampling

랜덤 오버 샘플링은 아주 간단한 방식입니다. 기존에 존재하는 소수의 클래스 (Minority)를 단순 복제하여 비율을 맞춰주는 것입니다. 단순 복제하여 분포는 변화하지 않습니다만, 숫자가 늘어나기 때문에 더 많은 가중치를 받게 되는 원리입니다.



ROS는 언뜻 보기에 단순히 같은 데이터를 복제시키는 것에 불과하니 성능이 좋지 않을 것 같지만, 실험적으로 유효할 때가 종종 있습니다. 그러나 똑같은 데이터가 증식되다보니 오버 피팅의 위험이 도사리고 있습니다.

```

from imblearn.over_sampling import RandomOverSampler, SMOTE, BorderlineSMOTE, ADASYN
from collections import Counter

# ROS(Random Over Sampler)

ros = RandomOverSampler(random_state = 42)
X_res, y_res = ros.fit_resample(X, y)

print('Resampled dataset shape %s' % Counter(y))
print('Resampled dataset shape %s' % Counter(y_res))

```

```

Resampled dataset shape Counter({2.0: 16968, 1.0: 6267, 0.0: 3222})
Resampled dataset shape Counter({1.0: 16968, 2.0: 16968, 0.0: 16968})

```

SMOTE(Sythetic Minority Over-Sampling Technique)

가장 유명한 SMOTE입니다. 원리는 그다지 어렵지 않습니다. 그러나 거의 모든 오버 샘플링 기법이 이 SMOTE를 뿌리로 뻗어나갔다고 해도 과언이 아닐 정도로 많이 쓰이는 기법입니다.

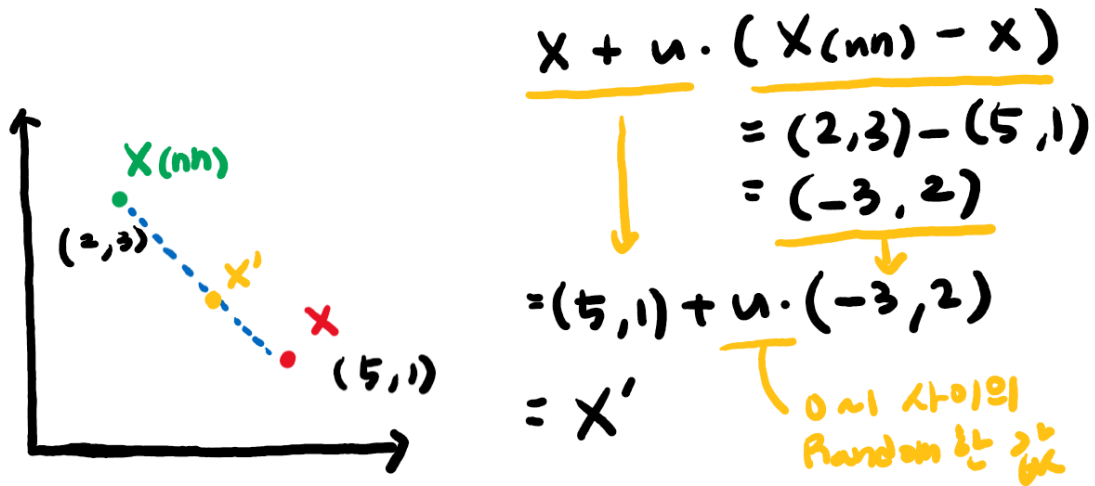
SMOTE는 간단히 말해서 임의의 소수 클래스 데이터로부터 인근 소수 클래스 사이에 새로운 데이터를 생성하는 것입니다. 무슨 말인지 자세히 살펴보겠습니다.

SMOTE는 먼저 임의의 소수 클래스에 해당하는 관측치 X 를 잡고, 그 X 로부터 가장 가까운 K 개의 이웃 $X(nn; \text{Nearest Neighbors})$ 를 찾습니다. 그리고 이 K 개의 $X(nn)$ 과 X 사이에 임의의 새로운 데이터 X' 를 생성하는 것입니다. 식은 아래와 같습니다.

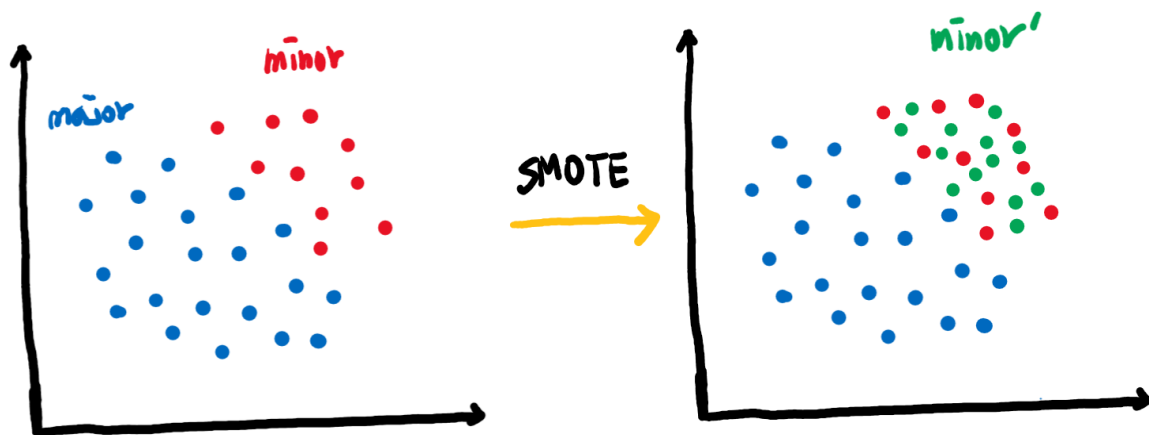
$$\text{Synthetic} = X + u \cdot (X_{(nn)} - X)$$

X 의 nearest neighbor
 u : 균등분포 (0,1)
 $X_{(nn)}$: 소수클래스 관측치

식만 보면 무슨 소리인가 싶겠지만, 아래 좌표와 식을 함께 보면서 해석을 해보겠습니다.



우리는 임의의 소수 클래스 관측치 $X(5, 1)$ 을 잡았습니다. 그리고 가장 근접한 이웃 K 개 중에서 임의로 $X_{(nn)}(2, 3)$ 을 잡았습니다. 우리는 여기서 $(X_{(nn)} - X)$ 의 값에 0~1의 값을 가지는 u 를 곱해서 기존의 X 에 더해줍니다. 그 결과가 임의의 새로운 데이터 좌표 X' 입니다. 한마디로 X 와 $X_{(nn)}$ 를 잇는 일직선 상에 임의의 데이터 포인트를 생성한다고 보시면 됩니다. 이 작업을 K 개의 다른 $X_{(nn)}$ 마다 진행합니다. 그 결과는 아래와 같습니다.



소수의 클래스 사이에 초록색의 새로운 데이터가 생겨난 것을 확인할 수 있습니다.

```
# SMOTE(Synthetic Minority Over-Sampling Technique)

smote = SMOTE(random_state = 42, k_neighbors = 5)
X_smote, y_smote = smote.fit_resample(X, y)

print('Resampled dataset shape %s' % Counter(y))
print('Resampled dataset shape %s' % Counter(y_smote))
```

```
Resampled dataset shape Counter({2.0: 16968, 1.0: 6267, 0.0: 3222})
Resampled dataset shape Counter({1.0: 16968, 2.0: 16968, 0.0: 16968})
```

5.6 마치며

이제 코드를 봅시다.

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
```

LinearDiscriminantAnalysis 는 클래스 간의 분리를 최대화하는 방향으로 구성된 선형 부분 공간에 입력 데이터를 투영하여 차원 축소를 수행하는데 사용할 수 있습니다. 출력의 차원은 반드시 클래스 수보다 적기 때문에 일반적으로 차원이 크게 감소하고 다중 클래스 설정에서만 의미가 있습니다. ⇒ LDA

```
from sklearn.naive_bayes import MultinomialNB
```

scikit-learn에 구현된 나이브 베이즈 분류기는

1. GaussianNB,
2. BernoulliNB,
3. MultinomialNB 이렇게 세 가지입니다.

GaussianNB는 연속적인 어떤 데이터에도 적용할 수 있고

BernoulliNB는 이진 데이터를

MultinomialNB는 카운트 데이터

(특성이 어떤 것을 헤아린 정수 카운트로, 예를 들면 문장에 나타난 단어의 횟수입니다)에 적

용됩니다
