o 분류분석

* 회귀 vs 분류
  + 회귀 : Y가 연속적
  + 분류 : Y가 순서 없는 범주값 ex) 차종, 인종, 증상
    - 순서가 있는 것도 없는 것처럼 처리할 수 있지만 성능이 떨어짐
* 분류를 하는 함수로 Sigmoid 채택(자연현상을 가장 잘 반영)
  + 일정 이하가 되면 0, 일정 이상이 되면 1, 미분 가능 등등의 조건
  + log(odd) => logit
* logit이 회귀로 표현된다고 해서 로지스틱 회귀

o 최소제곱

* 확률간의 에러(차이)의 제곱합
* 로지스틱의 회귀에서도 확률간의 제곱합 차이를 최소화하는 b0, b1을 찾음
* 문제 : 수학적으로 찾을 수가 없음. 수치적으로 풀어야하는 문제
* gradient decrease 방법으로 찾아야 함

o 최대 우도(최대로 ~할 것 같은 정도) ; ML

* 우도(Likelihood)란? : 어떤 모델을 가정했을때, 현재 보고 있는 데이터를 관측할 확률

=> 어떤 데이터를 관측했을 때, 그 데이터가 관측된 이유는 일반적으로 그 확률이 가장 높았기 때문이라고 통계적인 관점에서 생각됨 => 그 관측된 데이터가 나올 확률을 최대화 하는 P가 진짜 P다.

* 최대 우도를 계산하기 위해서는 로그값 취해서 미분값이 0인 것( 역 컨케이브 함수이므로)
* ML은 비선형 모델에서 선호된다.

o 로짓(log(odd))의 의미는?

* odd = p/1-p 는 p=0.5 기준으로 비대칭적임(0~1 / 1~ 무한대)

=> 비대칭적인 odd값을 -무한대 ~0 / 0~ +무한대로 맵핑시키기 위해(linear 값으로 맵핑되기 쉽게) 대표적으로 log 함수를 씀( 이전에는 맵핑시키기 위해 가우시안 분포를 썼음)

결론적으로, log를 취하는 이유는 맵핑을 하는 하나의 방법임.

* 로짓값이 0(기준값을 0.5로 잡았을때)이 되는 선(Decision boundary)을 기준으로 Y=0, Y=1 을 나누게 됨

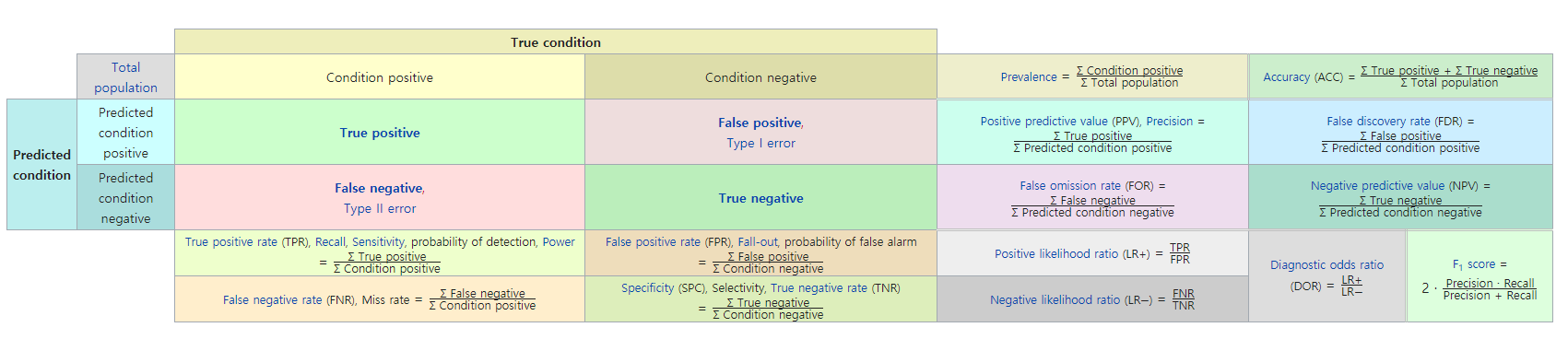
o 클래스가 여러개인 경우(Multi Bionomial)

o 판별 분석(DA)

* 선형판별분석(LDA)
  + 각 클래스들은 모두 정규분포를 따른다고 가정
  + 입력 변수가 연속형 변수일 때 분류에 적합.
  + **LDA 정규분포**를 따르는 확률 분포 f들이 평균은 다르지만 분산은 같다(가정)
  + 판별 함수 값이 가장 큰 클래스로 x를 추정한다.
  + 선형 판별 분석의 3가지 파라미터를 찾아야함(두 클래스의 각 평균, 분산)
  + 평균 => 클래스별 표본 평균, 분산 => 각 클래스의 분산의 평균 등
  + 로지스틱 회귀와 vs LDA
    - 공통 : Decision boundary 1개 존재(직선형 경계선), 클래스가 2개일땐 유사한 결과
    - 차이 : LDA의 경우 클래스가 많아지면 변수가 적기 때문에 용이함
    - LDA 문제점 : 정규분포가 아니라면 사용할 수 없음.

ex) 범주형 변수(더미 variable)의 경우 정규분포가 아니기 때문에 범주형 변수를 이용한 분류의 경우 불가능.

* 이차판별분석(QDA)
  + LDA와는 달리 QDA는 분산이 다른 것을 가정한다.
  + Decision bondary가 2개 => 경계가 직선이 아님(비선형 경계선)
  + 더 복잡한 모델링일 때 사용(평균도 다를 수 있고, 분산도 다를 수 있음)
  + 단, 복잡한 모델의 경우 test set이 입력됬을 때 좋을 수도 있고 아닐수도
* 이진판별 평가하기(모델 성능 평가)
  + 정확도 : accuracy ;
  + 오차율 : error rate ;
  + 생각해볼문제 : 특정 클래스가 지나치게 치우친 경우, 모델의 성능 평가에 있어 문제점이 있음. 모든 결과를 특정 클래스로 지정하더라도 성능은 마치 높은 것처럼 나오지만, 해당 모델은 아무런 쓸모가 없음(Ex, 공장 불량품 찾기)
* 찾고 싶은 클래스 : 양성(Positive)
* 그 반대 : 음성(Negative)



출처 : <https://en.wikipedia.org/wiki/Precision_and_recall>

* 동일한 지표에 대해서 부르는 이름이 도메인마다 다름
* ROC 커브 => 분류기 성능 평가
  + 문제 : ROC는 Scoring 할 수가 없음
    - => AUC : ROC 커브 아래 쪽의 영역 (0.5~1) 사이의 값을 가짐.
    - AUC : 1 완벽한 판별기
    - AUC : 0.5 랜덤 판별기
  + 판별 기준점은 어떻게 정해야 하는가? (i.e. p(x) = 0.5 기준)
    - 도메인에 따라 다름. 의학에서는 꽤 엄격한 판별기준

o Train set vs test set 나누기

* 나누는 방법1 : 시간순으로 일정 비율로 나누기(시간 흐름에 따른 변화가 있다고 생각될 경우)
* 나누는방법2 : 랜덤으로 나누기
  + 즉, 무조건 랜덤으로 뽑아서 일정비율로 나누는 것이 좋은 건 아님

o Feature-selection : **어떤 변수가 유용한** 변수인지 판별하는 방법

* 문제 : 보통 변수의 제곱에 해당되는 모델이 필요함(빅 데이터의 경우 너무 계산량이 많음)
* 방법 : stepwise selection
* 해결법 : 1:1 관계로 상관성을 가지고 유의미한 변수들을 추려낸 후, 적은 변수 개수를 가지고 Feature-selection을 함

o sckit-learn의 로지스틱회귀 vs 스텟 모델의 로지스틱 회귀

* sckit-learn의 로지스틱회귀 : 모델 성능을 높이기 위해 penelization 기법을 사용
* 스텟 모델의 로지스틱 회귀 : 순수 로지스틱 회귀 모델

o Cross-Validation(CV) 교차검증 (모델 성능 판단)

* **여러가지 모델이 있다면 어떤 것을 골라야 하는가?**
  + 방법1 : P-value 를 보고 유의미한 변수를 고를 순 있지만, 그렇다고 모델의 성능이 향상된다는 것은 아님(대략적으로는 일치하지만, 항상 그렇지는 않다)
  + test - set의 cost를 최소화하는 점을 트레이닝 셋에서 찾는 방법이 교차검증이라고 함.
  + 교차검증을 통해 나오는 에러의 최저값은 test set에서도 에러의 최저값이 될 것이라는 것이 증명되어 있음(샘플이 충분히 많아진다면)
  + 교차검증 에러는 진짜 에러가 아님(진짜 에러는 test-set임), 교차검증을 통해 최적의 값을 유추할 뿐.
  + 컴퓨터를 이용하여 반복작업하는 것

o 통계에서 만드는 tree모델과 컴퓨터학에서 만드는 tree모델은 같은 트리 모형을 쓰지만 다름

o 모델의 성능을 향상시키기 위해 취하는 방법

1. Feature Selection
2. Dimension Reduction (대표적인 것이 PCA)
3. Penalization(Sklearn => 파라미터에 대한 제한이 있는 로지스틱 회귀 모델)
   1. 왜 성능이 증가하는 것인가?

Q. 교차검증의 의미? 샘플수가 적을때 모든 샘플이 트레이닝셋으로 사용될때의 교차검증?

* 방법론이 여러개가 있음.
* 결론적으로, 트레이닝 셋 내에서 여러번 하는것이 중요한 의미임
* 모델을 평가하기 위해서 트레이닝셋을 가지고 여러번 반복하는 것임(테스트 셋은 절대 보이지 않는다는 전제니깐)

Q. FS와 CV의 선후

* 모델을 생성할때 FS를 쓰고, 생성된 모델을 평가할때 CV를 씀

Q. PCA 분석을 했을때 Y 예측값은 어떻게 변환? 분류가 아닌 예측일 경우.

* Y값은 변환시킬 필요는 없음