Data leaks

- An unexpected information in the data that allows us to make unrealistically good predictions.

- Exclusive to ML competition.

- Contents
    1. Leakage types and examples
    2. Leaderboard probing
    3. Concrete walkthroughs

Data leaks - 1. Leakage types and examples

(1) Leaks in a time series competition

- First thing to look : train/public split, is it on time?
  - 그렇지 않은 feature가 가장 중요한 정보가 될 것이다.
  - (Ex.) 내일 주가를 예측하기 위해 모레 주가를 활용하는 경우.
- Even when split by time, test set features may contain information about future.

(2) Unexpected information

- **Meta data**
  - tr/te 이외의 img, text데이터가 주어지는 경우, 메타 데이터(파일생성일자, 해상도 등).
  - (Ex.) 개/고양이 분류문제에서, 개/고양이를 찍은 카메라가 다른 경우.
  - (Ex.) "Truly Native?" competition : the dates from zip archives.
- **Information in IDs**
  - IDs may be a hash of something.
  - (Ex.) Caterpillar competition : tube id
- **Simply adding row number of relative number**
- **Row order**
  - (Ex.) rows next to each other usually have the same label ("TalkingData Mobile User Demographics" competition)

Data leaks - 2. Leaderboard probing

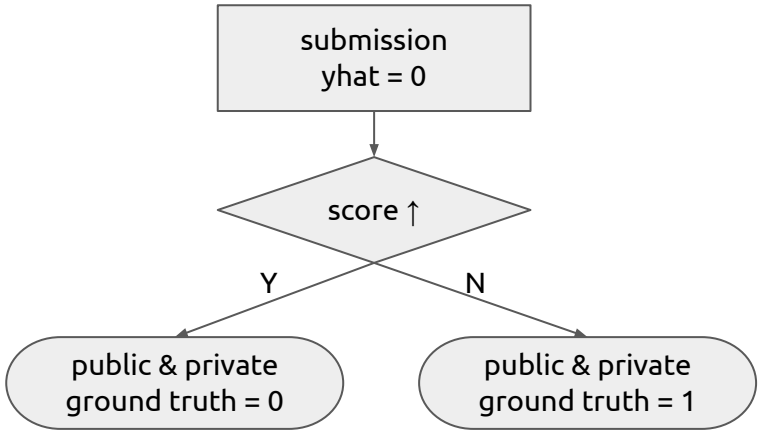
LB probing에는 두 종류가 있다.

- **[Type 1] 예측값 몇 줄만 바꿔서 최대한의 Public LB 정보 알아내기**
  - Oleg Trott 포스트 참고: 최소한의 submission으로 알아내는 방법.

- **[Type 2 ★] Consistent categories connected to 'id' are vulnerable to LB probing**
  - 여러 번의 submission으로 private testset 정보를 알아오는 전략.
  - (Ex.) Company of user in RedHat competition, Year, Month, Week in WestNile competition

id	...	y
1	...	0
1	...	0
1	...	0
2	...	1
2	...	1
2	...	1

Private
Public



Data leaks - 2. Leaderboard probing

- **[Type 2 ★] Consistent categories connected to 'id' are vulnerable to LB probing**
  - public/private 모두 target label의 분포가 동일할 수 있다.
  - (Ex.) Quora Question Pairs competition (binary classification)
    - tr/te 분포는 달랐지만 public/private 분포는 같았다.
    - logarithmic loss 계산식을 정리하면
      - C : constant predictions (predicted probability of outcome)
      - N : real number of public testset rows
      - N1 : target=1인 public testset row수.
      - L : public LB score
    - public testset에서 정답이 1인 실제 비율(=50%)을 알 수 있다.

Adapting global mean via LB probing:

$$-L*N=\sum_{i=1}^{N}(y_i \ln C+(1-y_i)\ln(1-C))$$

$$-L*N=N_1 \ln C+(N-N_1)\ln(1-C))$$

$$\frac{N_1}{N}=\frac{-L-\ln(1-C)}{\ln C-\ln(1-C)}$$

  - 이를 통해, trainset이 testset과 동일한 target 분포를 갖도록 만들 수 있다.

Please note: as an anti-cheating measure, Kaggle has supplemented the test set with computer-generated question pairs. Those rows do not come from Quora, and are not counted in the scoring. All of the questions in the training set are genuine examples from Quora.

Data leaks - Peculiar examples

(1) Truly Native
- (주제) Predict whether the content in an HTML file is sponsored or not
- (leaks) archive date
- date를 제외하더라도, HTML Text안에 여전히 날짜와 관련된 정보가 있다. (news)

(2) Expedia Hotel Recommendations
- (주제) 사용자 로그 정보로, 어떤 호텔 그룹을 예약할지 예측하는 것.
- (leaks) distance가 leak이었다. 좌표를 이용해서 ...

(3) Flavours of physics
- (주제) whether the signal was artificially simulated.
- (leaks) 하지만 시뮬레이션은 완벽할 수 없기에, 리버스 엔지니어링을 통해 완벽한 점수를 얻을 수 있었다.

(4) Pairwise tasks
- (주제) Can you identify question pairs that have the same intent?
- (leaks) 모든 Pairwise task의 공통적인 특징 :
- 모든 조합에 답하지 않도록 한다.
- = 그러다보니 항상 랜덤하지 않게 샘플링 된다.
- = 빈번한 아이템일수록 중복해서 등장할 가능성이 크다.
- N x N 인접행렬을 만들어 벡터간의 유사성을 계산했다. 비슷한 이웃 집합을 갖고 있으면, 중복일 가능성이 크다.

Data leaks - Expedia challenge examples

사용자 검색내용, 클릭, 예약, 여행패키지 해당유무 등의 정보를 바탕으로,
어떤 "**hotel_cluster**"를 예약할지 예측해보자.

leak!

Column	Description	Data Type
user_location_city		int
orig_destination_distance	사용자의 도시 ~ 실제 예약한 호텔의 거리	Double
is_package	1 : if the click/booking was generated as a part of a package (i.e. combined with a flight) 0 : otherwise	int
hotel_cluster (*)	가격, 사용자 평점, 도시 중심으로부터의 거리 등의 특징이 비슷한 호텔들의 집합	int

Data leaks - Expedia challenge examples

testset에는 이미 trainset에서 본 distance pair가 있었고, 그걸 일종의 label로 삼았다.
그래도 절반은 처음보는 pair였고, 이걸 처리하는 데 2가지 방법을 사용했다.

1. Features based on counts on corteges of such nature

또 다른 pair들의 count를 만들었다 : 사용자의 도시, 호텔 국가, 호텔 도시에 특정 호텔 클러스터가 몇 개나 있는지
그 변수들로 머신러닝 모델을 만들었다.

2. Try to find the true coordinates of user cities and hotel cities.

좌표로 destination distance 변수를 유추할 수 있을 것이라 예상했음.

Data leaks - Expedia challenge examples

2-(1) 거리 구하는 방법 "Haversine formula"

정확한 세 점의 좌표를 알고, 위치를 모르는 점으로부터 세 점들 사이의 거리를 알면, 모르던 좌표를 알 수 있다.

그래서 일단 리버스 엔지니어링으로 대도시 세 곳의 좌표를 찾고, 더 많은 도시의 좌표를 찾아갔다.

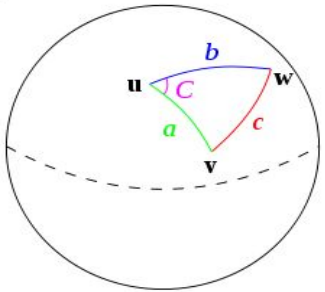
하지만 rounding이 반복되면서 부정확해졌다. 바다 위에도 도시가 있다고 계산되는 오류도 있었다.

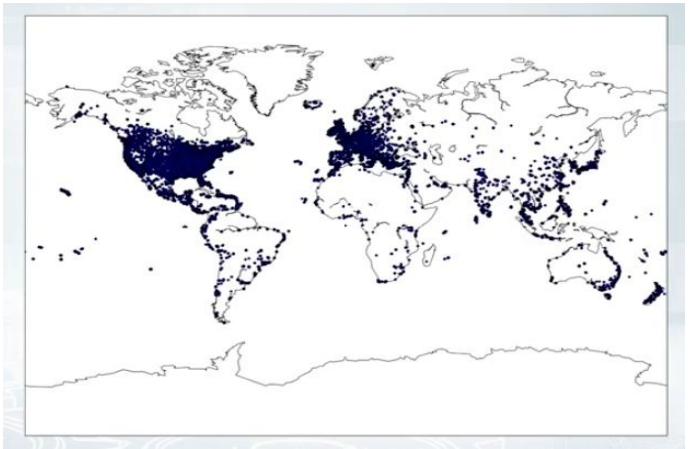
2-(2) New Version.

방정식 3개에서 확장해서, 모든 거리와 정확한 좌표들을 바탕으로 더 거대한 system of equations를 만들었다.

(scipy를 써서 sparse matrix를 효율적으로 계산했다.)

$$d = 2r \arcsin\left(\sqrt{\operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)}\right)$$
$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$





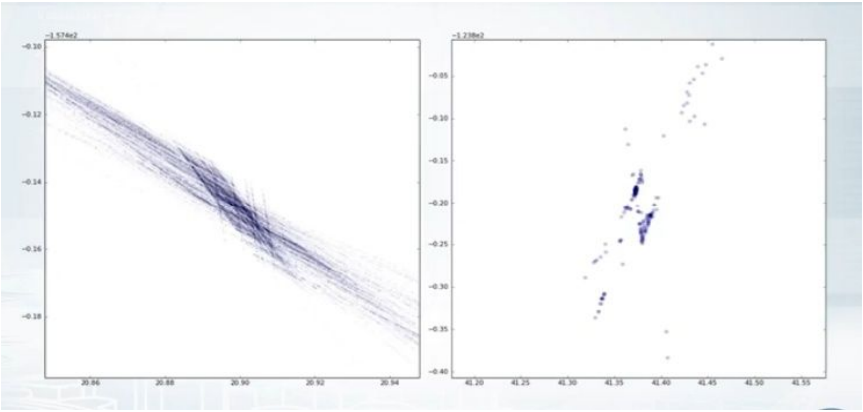
Data leaks - Expedia challenge examples

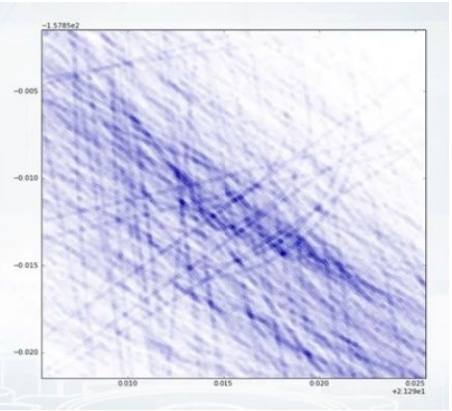
Feature Engineering

모든 사용자 도시 ~ 특정 호텔까지를 반경으로 하는 원을 그려보자.

그런데 하나의 도시에 존재할 수 있는 호텔은 한정적이기 때문에, 호텔은 원들의 교차점에 있게 되고
더 많은 원들이 그 지점을 지나칠수록 호텔이 그 지점에 있을 확률은 더 높아진다.

하나의 점이라고 콕 찝어 말할 순 없지만, 교차점들의 군집이 보이지.
이걸 바탕으로 정보를 구체화해보면 ...





Data leaks - Expedia challenge examples

Feature Engineering

도시마다, 도시의 중심을 둘러싸는 격자를 그려보자.

이제 학습데이터를 이용해서, 격자 한 칸마다 어떤 타입의 호텔이 몇 개나 있는지 count해보자.

만약에 격자를 지나는 원이 있으면, 그 원이 어떤 호텔 타입과 매핑된다고 +1 count 하는거지.

그 counter들의 sum, avg, max등을 변수로 가공했다.