배깅의 목적은 바이언스를 그대로 분산을 낮추는

중요한 피쳐가 하나 있을수록

트리 모양이 비슷해지면, 분산은 의미가 없음

각각이 독립적이어야함.

그러나 트리의 모양이 비슷하면 트리끼리 독립적이지 않는다.

스플릿한 피쳐를 랜덤하게 선택하니까 트리 모양이 달라질 것

트리 간의 코레레이션을 줄이는 것이 랜덤 포레스트임.

앙상블에서 랜덤 포르스트나 배깅은

부트스트래핑해서 데이터 셋을 각각 만들고

분류기를 각각 데이터 셋에 학습 시킴-> 따라서 서로가 독립적으로 학습하니까, 병렬처리가 가능 하다(코어)

-앙상블(부스팅)

부스팅은 병렬 처리가 가능 하지 않다.

스테이즈와이즈 어디티브 모델링:여러 모델을 한꺼번에 쓸 때, 사용할 수 있는 알고리즘임!

스테이즈와이즈 어디티브 모델링은 부스팅의 기본 컨셉임 (스테이즈 와이즈는 스텝\_즉 순서가 있음)

스태킹은 앙상블의 앙상블임

* 위크 러너는 프리딕션이 굉장히 낮음
* 머신 러닝의 최종 목적은 일반화임

모델의 복잡도가 늘어나면(예측력이 높은 모델은) 높은 베리언스가 발생함

* 그래서 복잡도가 떨어지고(예측력이 낮고) 낮은 베리언스
* 따라서 높은 바이어스 발생

높은 바이어스를 잡기 위해 위크 러너를 합쳐보자

데이터 셋 부분 별로 각각의 위크 러너 적용

배깅은 병렬적으로 하나의 트리 모형이 여러 개 있는 것(완벽한 트리 모형)

부스팅에서 중요한 점은 위크 러너를 (리 웨이트 된)

똑 같은 데이터 셋 위에서 모델을 학습 시키면 상관관계가 높아지지만-> 위크러너를 사용함으로서 바이어스를 잡아 보자

리웨이트 되었따는 것은-> 틀린 샘플에 대해 클래스 웨이트를 높게 주는 것, 그럼 틀린 샘픔이 틀리면 로스 펑션이 올라가니까 잘 분류하려고 함.

ㅅ -> ㅅ+1 : ㅅ+1 번째 데이터는 가중치가 높은 샘플이 뽑힐 확률이 높아지는 정책 사용.

시퀀셜하게 학습한다!

언커버드 영역을 찾아서 학습한다!

아다 부스트는 위크 러너가 디시젼 스텀프(뎁스가 1인 것)

* 아다 부스트 동작 원리

: 모델 자체가 언커버된 부분이 시퀀시얼 하게 합치는 것

계속 부족한 부분을 메꿔 주는 것임(더하는 것임)

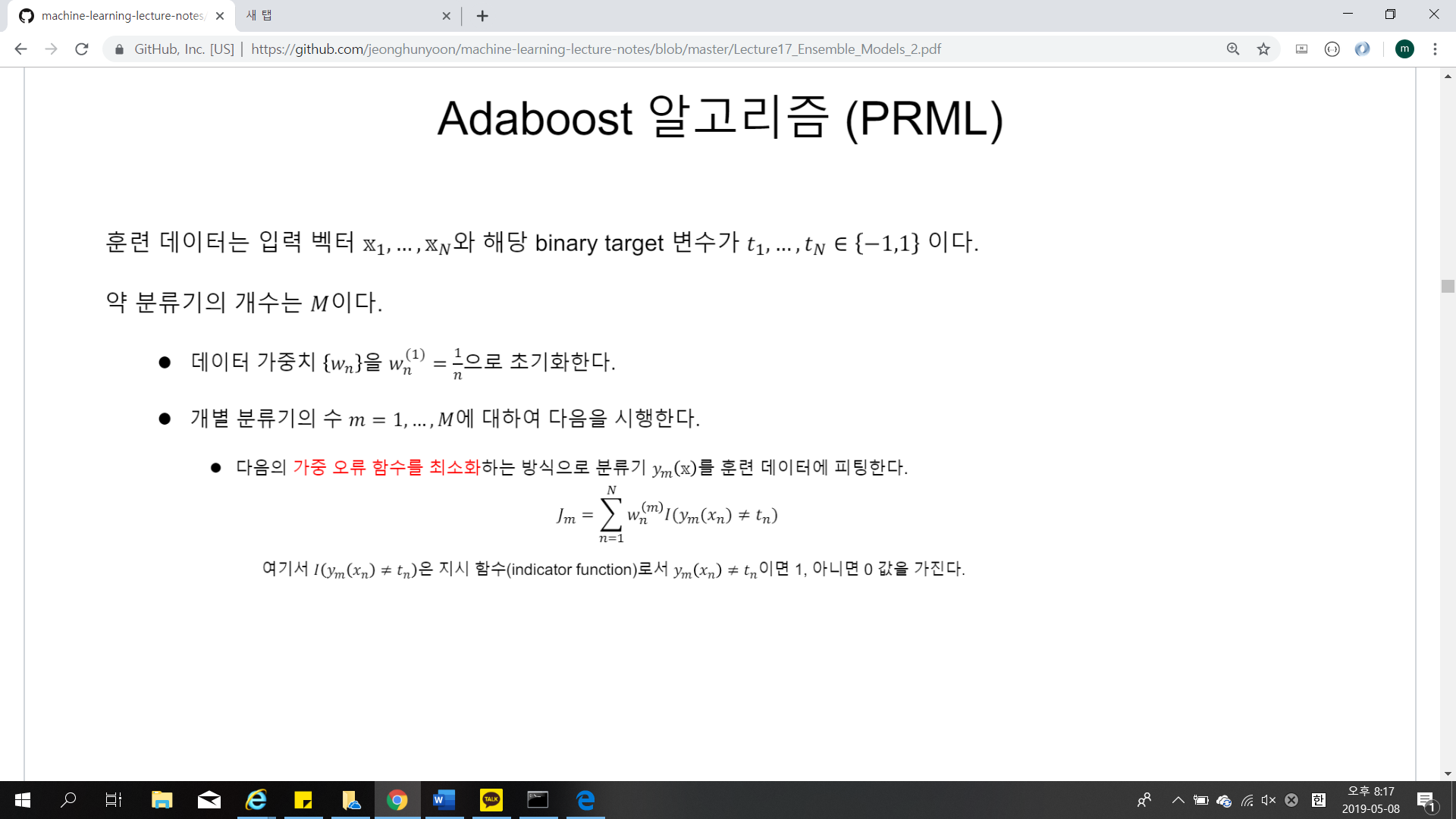
더하는 것이라는 게 메져리티 보팅 하는 것이 아니라,

부족한 부분을 계속 보충해가며 더해가는 것임

* 아다 부스트 알고리즘

1. 전체 데이터에 동일한 가중치를 둠(첫번째는 가중치 없이 디시젼 트리)
2. 다음부터 잘못 분류된 샘플을 찾아서 그 가중치를 더하면 됨.
3. 잘못 분류한 데이터의 샘플(가충치)를 구해서 거기에서 1을 뺴면 해당 분류기의 신뢰도 가 됨( 내가 맞춘 놈의 가중치가 그 분류기의 신뢰도임)
4. 신뢰도를 기반으로 다음에 잘못된 데이터에 가중치를 계산 한다.

트리 모델에서 디시젼 스텀프를 싸인함수로 많이 표현함 (1아니면 -1)



* 분류기에서 측정한 값과 실제 값이 다른 경우 웨이트를 늘린다.

웨이트를 크게하는 것은 데이터 가중치(웨이트 엔) 임

학습하기 전에 그 다음 단계에서 잘못된 것들을 모아서 그 웨이트를 낮추는 것

(그레디언트 디선트)

로스 펑션을 만들고 나서 피팅을 할 때, 로스 펑션상 밸류 값 높은 것들을 모아서 그 다음에서 잘못 분류된 것의 웨이트를 낮춰야 함.

* 로스 펑션은 오 분류 율임.
* t, t+1 이라고 할 때, t+1에 t에서 잘못 분류된 데이터 들에 대한 웨이트를 높여서(로스 평선 상) 잘못 된 데이터 들이 잘 분류 되게 하고 최종 적으로 그 전체 웨이트를 낮춰 준다!

그래디언트 부스팅은 함수랑 파라미터이 학습을 통해 변함

오류율이 미분 가능한 손실 함수가 되는 것임.

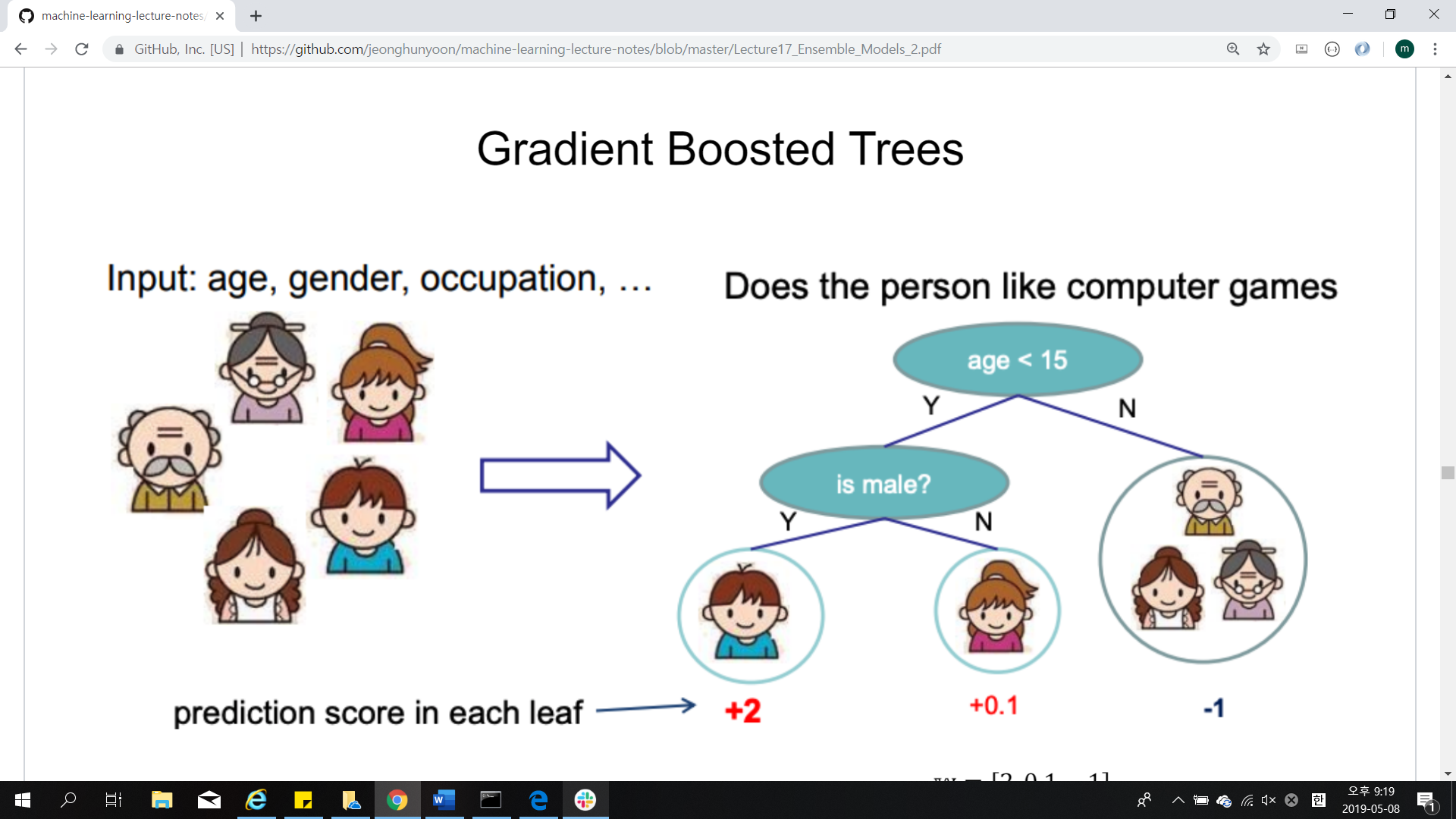
오차: 모집단

잔차: 표본 집단

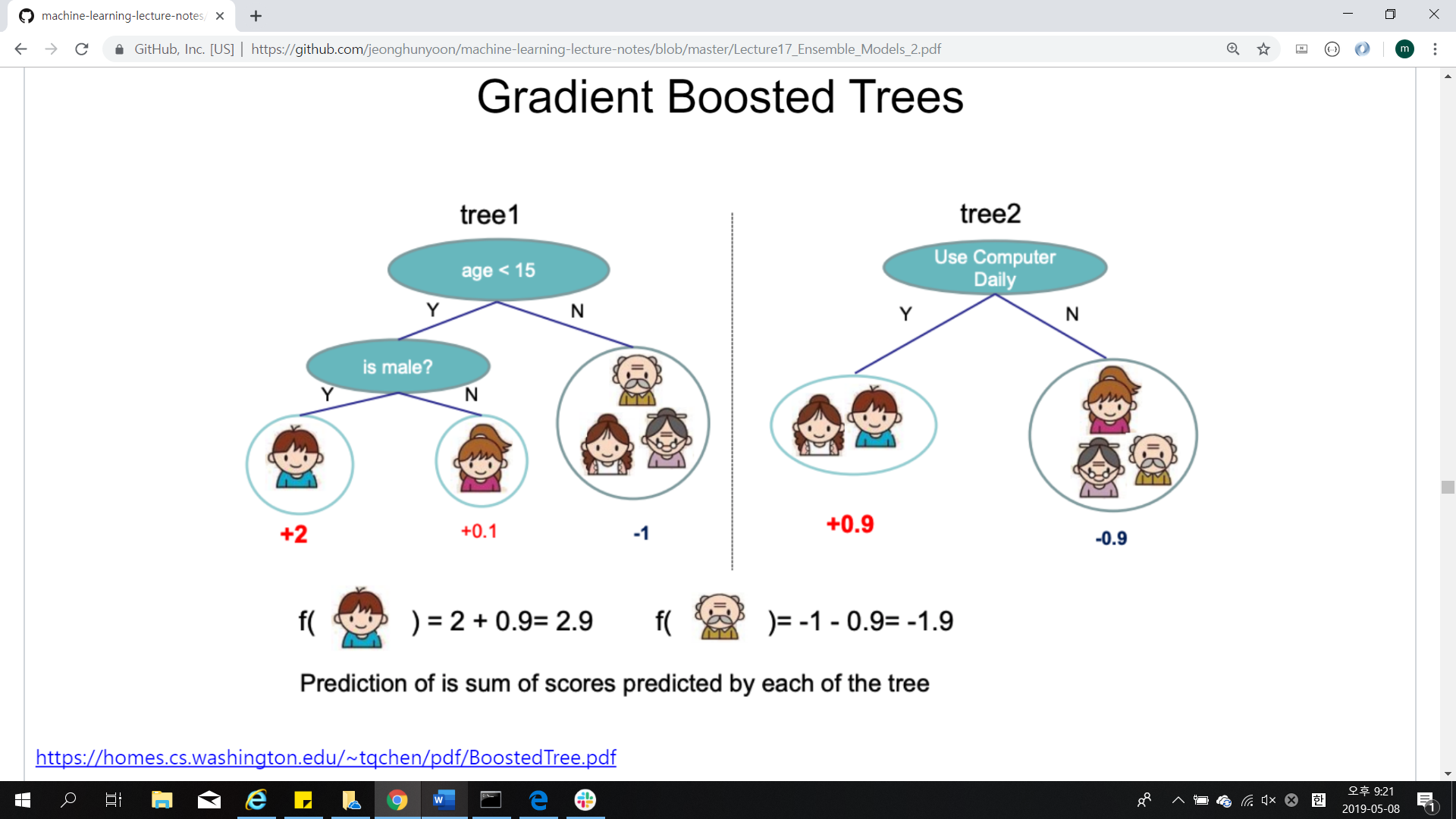
미분 가능한 함수 :

그레디언트 부스팅에서 오류율을 오차항임(잔차)

* 오차항을 타겟으로 그레디언트 디센트를 쓰는 이유는?



여기서 중요한 것은 각 리프에 가중치를 더하는 것이고 계속 에드 한다는 것임



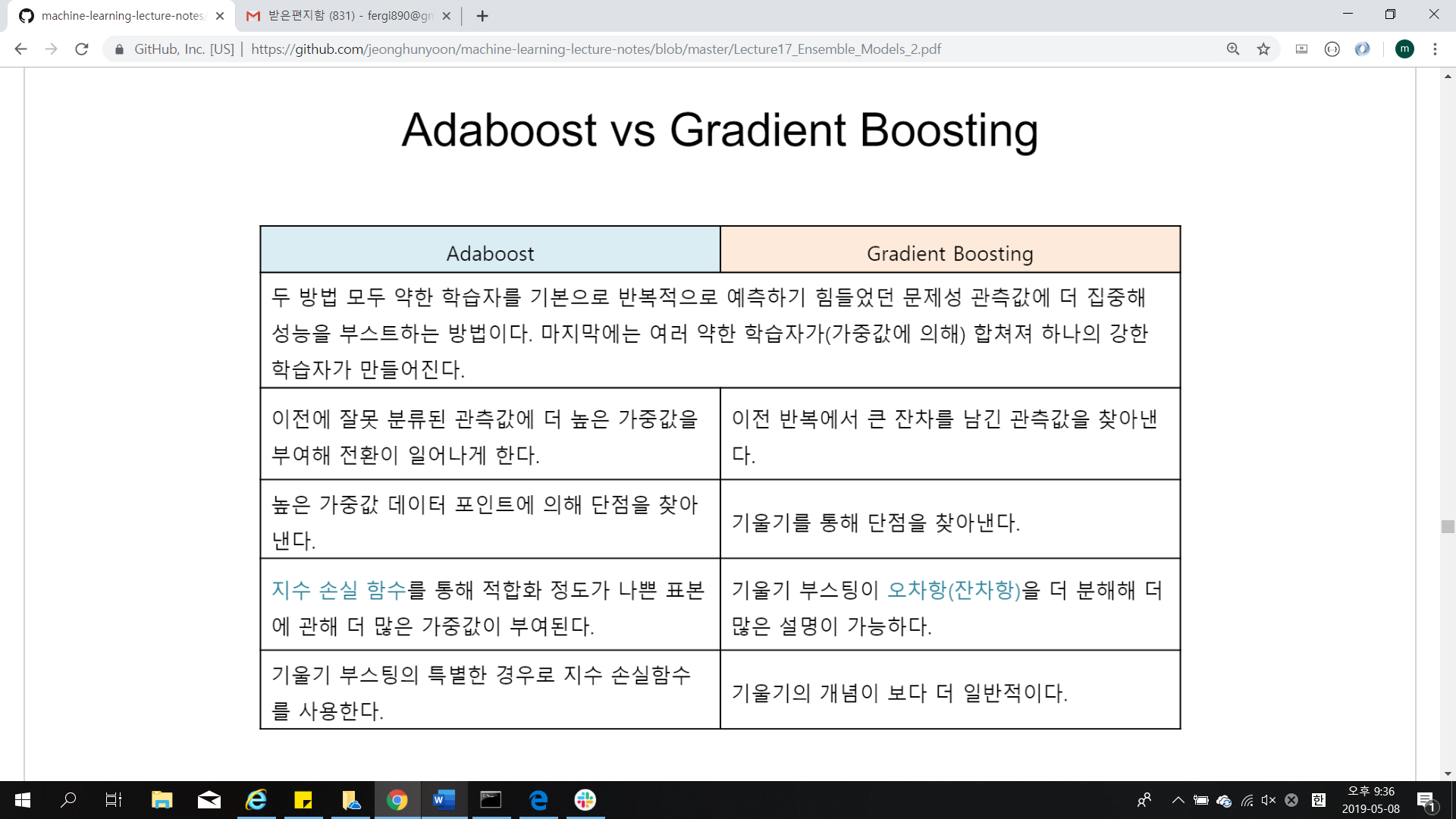
가중치를 계속 더하는 것

* 디시젼 트리와 랜덤 포레스트 그리고 그레디언트 부스티드와 다른 점

분기할 때, 디시션 트리나 랜덤 포레스트는 지니 계수나 엔트로피를 사용하나

그레디언트 부스트는 로스펑션이 최소가 되는 값에서 분기 하는 것임.

그래서 그레디언트 부스트는 수렴하는 값이면서 가장 가까운 값임!



아다부스트는 로직스틱 리그레션의 애디티브 펑션이다.

머신 러닝과 통계의 가장 큰 차이는 머신 러닝은 옵티마이제이션

그래디언트 부스팅을 얼마나 빠르게/팬시하게 만들까

엑스 쥐 부스트는 시스템을 최적화한 부분\_ 그래디언트 부스팅의 경우, 시간이 많이 소요됨. 따라서 엑스쥐부스트가 나옴 엑스쥐 부스트가 그래디언트 부스팅의 최적화 임!

<https://web.stanford.edu/~hastie/TALKS/boost.pdf->> 이거 꼭 봐!

<http://cs229.stanford.edu/notes/cs229-notes-ensemble.pdf->>이건 진짜 꼭봐

* 새미와 새미 알의 차이 점은 가중치에 있음!