깃허브 저장 루틴

1. 깃허브 연결 폴더에 파일/폴더 추가
2. 깃허브 실행
3. Fetch, push origin
4. Settings -> pages -> 외부에서도 접속 가능한 링크
5. 뒤에 접속 주소 붙여넣기
6. Html ipynb 차이

회귀: k=n일 때, 가장 가까운 n개의 레이블의 평균

내가 feature를 n개 쓴다=>학습시킴=>최적의 가중치 값 결정 (결정됨=학습 완료됨)

회귀 모델

: MSE를 최소화하는 W(가중치)와 BIAS 값을 나타낸 모델 (as 선형방정식)

: FEATURE가 하나일 때는 직선, 2개일 때는 평면, 높은 차원에서는 초평면

선형 회귀에서 결정계수: 회귀 모델에서 예측의 적합도 측정

결정 계수: 사이킷런에서 score로 표현

하이퍼 파라미터 = 모델이 학습하는 것 x, 사용자가 직접 설정해줘야하는 것

\*모델 파라미터 = w1, w2

평균제곱오차 MSE: (실제값-예측값의 제곱)의 평균

MAE: (예측값-예측값 햇)의 평균

RMSE: MSE의 제곱근 값

RMLSE: 예측값의 로그

선형 모델

라쏘 릿지 공통점2: 가중치의 절대값을 0에 가깝도록 규제

라쏘 릿지 공통점: 과대적합한 것을 덜 적합하게 규제

라쏘 회귀 L1 : 0으로 만들어버리는 애가 있어서 변수 줆

0이 되는 값 생김

릿지 회귀 L2 : 0이 되는 가중치 없기 때문에 변수 개수 변동 x

변수가 다 살아남음

Alpha 규제를 통해 선형 회귀 모델의 과대적합이 줄어든다

Alpha 값이 클수록 규제 많음

결정트리

입력 데이터가 있을 때 변수에 따라 각각의 노드를 지날 때마다 조건에 해당되는 최적의 기준으로 모델을 나누고, 새로운 데이터를 만들어진 조건에 따라 내려가면 되는 것

분류, 회귀 문제에 풀 수 있음

[머신러닝]

1. 지도 학습 -레이블 있음

-회귀 (Regression)

Knn

Linear Regression

-선형 함수로 만들어짐

-학습이 완료되면 MSE 값이 최소가 되는 w, b 값이 정해진다

-feature가 많아지면 많아질수록 과대적합 발생 가능

Lasso(L1)

Ridge(L2)

Lasso와 Ridge의 공통점: w의 규제를 통해 0에 가깝게

차이점: Lasso는 0으로 만듦

Ridge는 w값이 0이 되지 않음

1. 비지도 학습 -레이블 없음

엔트로피(공식에 대입하여) =0이면 분기 x

분기하지 않는 노드 = if 노드

우리가 끊어주지 않으면 엔트로피가 0이 될 때까지 끝까지 나옴

Leaf 노드는 레이블을 예측해야함 (수치형일 경우 타겟값의 평균값)

과대적합을 줄이기 위해서 max depth 사용

앙상블 기법-랜덤 포레스트(성격이 다른 다수의 트리 이용)

1. 서로 다른 데이터에 대해 중복 허용해 과대적합 감소시킴
2. 무작위 일부 특성(feature) 선택 (30개 변수라면 루트 씌워서 매 트리마다 사용하는 변수는 5-6개)

학습에 사용하는 데이터로 트리를 만드는데, 데이터를 다르게 만들어서 중복 허용해 여러 개의 트리를 만든다 (서로 다른 데이터 만듦)

-가장 많이 예측된 값으로 최종 결과 예측

-수치형) 각 모델이 예측한 수치들의 평균 내어 최종 결과 보여줌

\*가지 치기 두 가지 방식 : 사전, 사후

의사결정트리(사전 가지치기 방식 사용)