5️⃣

05 _ 회귀

1. 회귀 소개

회귀 : 데이터 값이 평균과 같은 일정한 값으로돌아가려는 경향을 이용한 통계적 기법

여러개의 독립변수 -> 1개의 종속변수

이 때 독립변수 개수가 1개 : 단일회귀, 여러개 : 다중 회귀

회귀 계수(Regression coefficients) : 독립변수가 한 단위 변화함에 따라 종속변수에 미치는 영향력 크기

이 때 회귀계수가 선형 : 선형회귀, 비선형 : 비선형 회귀

- 대표적인 선형 회귀 모델

1. 일반 선형 회귀 : 예측값과 실제 값의 RSS(Residual Sum of Squares)를 최소화할 수 있도록 회귀 계수를 최적화하며, 규제를 적용하지 않은 모델
2. 릿지(Ridge) : 선형 회귀에 L2 규제를 추가한 회귀 모델. 상대적으로 큰 회귀 계수 값의 예측 영향도를 감소시키기 위해서 회귀 계수값을 더 작게 만드는 규제 모델
3. 라쏘(Lasso) : 선형 회귀에 L1 규제를 적용한 방식. 예측 영향력이 작은 피처의 회귀 계수를 0으로 만들어 회귀 예측 시 피처가 선택되지 않게 하는 것 (피처 선택 기능)
4. 엘라스틱넷(ElasticNet) : L2, L1 규제를 함께 결합한 모델. 주로 피처가 많은 데이터 세트에서 적용되고 L1 규제로 피처의 개수를 줄이고 L2 규제로 계수 값의 크기를 조정한다.
5. 로지스틱 회귀(Logistic Regression) : 강력한 분류 알고리즘. 일반적으로 이진 분류뿐만 아니라 희소 영역의 분류에서 뛰어난 예측 성능을 보인다. ex) 텍스트 분류

2. 단순 선형 회귀를 통한 회귀 이해

단순 선형 회귀 - 독립변수, 종속변수가 하나인 선형 회귀

최적의 단순 선형 회귀 모델을 만든다는 것은 실제 값과 회귀 모델의 차이에 따른 오류, 즉, 남은 오류(잔차)를 최소화 하는 것이다.

오류 합 계산시에는 Mean Absolute Error나 Residual Sum of Square을 사용.

오류 값을 구하는 방식

- 절댓값을 취해서 더함 (Mean Absolute Error)
- 오류 값의 제곱을 구해서 더함 (Residual Sum of Square)

→*Error2=RSS*

→ RSS를 최소로 하는 기울기, 절편 즉, 회귀 계수를 학습을 통해서 찾는 것이 머신러닝 기반 회귀의 핵심 사항

- 회귀식의 독립변수, 종속변수가 중심 변수가 아닌 w변수(회귀 계수)가 중심 변수임을 인지하는 것이 매우 중요!

$$RSS(w_0, w_1) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + w_1 * x_i))^2$$

(i는 1부터 학습 데이터의 총 건수 N까지)

- 회귀에서의 비용(Cost)이며, 회귀 변수로 구성되는 RSS를 비용 함수라고 함
- 머신러닝 회귀 알고리즘: 비용함수(=손실함수)를 지속해서 감소시키고 더이상 감소하지 않는 최소의 오류 값을 구함

3. 비용 최소화하기 - 경사 하강법

경사 하강법 : 점진적으로 반복적인 계산을 통해 W 파라미터 값을 업데이터 하면서 오류 값이 최소가 되는 W 파라미터를 구하는 방식

반복적으로 비용 함수의 반환 값, 즉 예측값과 실제 값의차 이가 작아지는 방향성을 가지고 W 파라미터를 지속적으로 보정해나감 → 오류 값이 더 이상 작아지지 않으면 그 오류 값을 최소 비용으로 판단하고 그때의 W 값을 최적의 파라미터로 반환

💡 **정리**

1. w1, w0를 임의의 값으로 설정하고 첫 비용 함수의 값을 계산
2. w1, w0의 값을 위의 편미분 값으로 업데이트 해주고 다시 비용 함수의 값을 계산
3. 비용 함수의 값이 감소했으면 다시 2번 반복, 더 이상 비용 함수의 값이 감소하지 않으면 그때의 w1, w0를 구하고 반복을 중지

4. 사이킷런 LinearRegression을 이용한 보스턴 주택 가격 예측

LinearRegression 클래스 : 예측값과 실제 값의 RSS를 최소화해 OLS 추정 방식으로 구현한 클래스

- 입력 파라미터
  - fit_intercept : 불린 값, 절편을 계산할 지 말지 결정
  - normalize : 불린 값, 회귀 수행 전, 데이터 세트를 정규화 할지
- 속성
  - coef : fit() 메서드를 수행했을 때 회귀 계수가 배열 형태로 저장하는 속성. Shape는 (Target 값 개수, 피처 개수)
  - intercept_ : 절편 값
  -

다중 공산성

- 피처 간의 상관관계가 매우 높은 경우 분산이 매우 커져서 오류에 민감해지는 현상
  - 독립적이고 중요한 피처만 남기고 제거하거나 규제를 적용하면 됨 + PCA(주성분 분석)을 통해 차원 축소도 고려

5. 다항 회귀 이해

- 다항 회귀 : 회귀가 독립변수의 단항식이 아닌, 2차, 3차 방정식과 같은 다항식으로 표현되는 것
- 사이킷 런의 PolynomialFeatures 클래스를 통해서 다항식 피처로 변환
  - fit(), transform() 메서드를 통해 변환 작업 수행

다항 회귀를 이용한 과소적합 및 과적합 이해

다항 회귀는 복잡한 다항 관계를 모델링할 수 있음. 단, 차수를 높일수록 학습 데이터에만 너무 맞춘 학습이 이루어져 정작 테스트 데이터 환경에서는 오히려 예측 정확도가 떨어짐 → 과적합 문제 발생

- Degree 1 : 단순한 직선으로서 학습 데 | 터의 패턴을 제대로 반영하지 못한다. (과소적합)
- Degree 4 : 실제 데이터 세트와 유사하다. MSE값도 가장 낮은 것을 확인할 수 있다.
- Degree 15 : MSE 값이 말도 안되게 큰 수치를 기록, 변동 잡음까지 지나치게 반영하여 테스트 값의 실제 곡선과는 완전히 다른 곡선이 만들어졌다. (과적합)

편향 - 분산 트레이드 오프

머신러닝이 극복해야 할 가장 중요한 이슈 중의 하나로 위의 예시에서 Degree 1인 경우에는 지나치게 한 방향으로 치우친 고편향(High Bias) 모델, Degree 15의 경우 지나치게 높은 변동성을 가지는 고분산(High Variance) 모델로 볼 수 있음.

일반적으로 편향과 분산은 한 쪽이 높으면 한 쪽이 낮아지는 경향.

높은 편향 / 낮은 분산에서 과소적합되기 쉬우며 낮은 편향 / 높은 분산에서 과적합 되기가 쉬움

→ 편향과 분산이 서로 트레이드 오프를 이루면서 오류 Cost가 최소가 되는 모델을 구축하는 것이 가장 효율적인 머신러닝 예측 모델을 만드는 방법. (전체 오류가 가장 낮아지는 '골디락스' 지점)

6. 규제 선형 모델 - 릿지, 라쏘, 엘라스틱 넷

규제 선형 모델 개요

규제의 필요성 : 회귀 모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 함.

- 비용 함수는 학습 데이터의 잔차 오류 값을 최소로 하는 RSS 최소화 방법과 과적합을 방지하기 위해 회귀 계수 값이 커지지 않도록 하는 방법이 서로 균형을 이루어야 함.

릿지 회귀

- Ridge : 선형 회귀에 L2 규제를 추가한 회귀 모델, 상대적으로 큰 회귀 계수 값의 예측 영향도를 감소시키기 위해 회귀 계수 값을 더 작게 만드는 방식
- 회귀 계수를 0으로 만들고 있지는 않지만, alpha가 커질수록 회귀 계수가 지속적으로 작아지는 것을 확인할 수 있음 → 회귀 계수를 0으로 만들지는 않음

라쏘 회귀

- 라쏘 : L1 규제 추가한 회귀 모델, 예측 영향력이 작은 피처의 회귀 계수를 0으로 만들어 회귀 예측시 피처가 선택되지 않도록 하는 것
- 불필요한 회귀 계수를 급격하게 감소시켜 0으로 만드는 것

엘라스틱넷 회귀

- 엘라스틱넷 : L1 + L2를 결합, L1 규제로 피처의 개수를 줄임과 동시에, L2로 계수 값의 크기 조정
- 라쏘 회귀가 서로 상관관계가 높은 피처들 중 중요 피처만을 선택하고, 다른 피처들의 회귀 계수를 0으로 만드는 성향을 가져, alpha값에 따라 회귀 계수 값이 급격히 변동할 수 있다는 단점 완화하기 위함.
- alpha: (a+b)값. a: L1 규제 alpha값 / b: L2 규제 alpha 값
- l1_ratio = a/(a+b)

7. 로지스틱 회귀

- 로지스틱 회귀 : 선형 회귀 방식을 분류에 적용한 알고리즘
- 시그모이드 함수 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류 결정
- x값이 아무리 커지거나 작아져도 y값은 0과 1사이 값 반환

8. 회귀 트리

- 트리 기반의 회귀 : 회귀 트리 이용 → 회귀를 위한 트리를 생성하고 이를 기반으로 회귀 예측을 하는 것

  분류 트리와의 차이점 : 리프 노드에 속한 데이터 값의 평균값을 구해 회귀 예측값을 계산

- 결정 트리, 랜덤 포레스트, GBM, XGBoost, LightGBM 등 모든 트리 기반 알고리즘은 회귀도 가능 → 트리 생성이 CART 알고리즘에 기반하기 때문
- 회귀 트리는 선형 회귀와 다르게 분할되는 데이터 지점에 따라 브랜치를 만들면서 계단 형태로 회귀선을 만듦.