파머완 5장

1. 회귀

- 회귀(regression): 사람의 키는 평균 키로 회귀하려는 경향을 가진다는 자연의 법칙이 있다는 것입니다. 회귀 분석은 이처럼 데이터 값이 평균과 같은 일정한 값으로 돌아가려는 경향을 이용한 통계학 기법
- 머신러닝에서의 회귀: 여러 개의 독립변수에 따라 아파트 가격이라는 종속변수가 어떤 관계를 나타내는지를 모델링하고 예측하는 것
- 회귀 계수: 독립 변수의 값에 영향을 미치는 것. 머신러닝 회귀 예측의 핵심은 주어진 피터와 결정 값 데이터 기반에서 학습을 통해 최적의 회귀 계수를 찾아내는 것을 말함.

회귀 계수의 선형/비선형 여부, 독립변수의 개수, 종속변수의 개수에 따라 여러 가지 유형으로
나눌 수 있습니다.

회귀에서 가장 중요한 것은 바로 회귀 계수입니다. 이 회귀 계수가 선형이나 아니냐,
에 따라 선형 회귀와 비선형 회귀로 나눌 수 있습니다.

그리고 독립변수의 개수가 한 개인지 여러 개인지에 따라 단일 회귀, 다중 회귀로 나뉩니다.

*제일 많이 사용하는 것은 선형 회귀. 실제 값과 예측값의 차이를 회소화하는 직선형 회귀선을 최적화하는 방식을 말함.

*지도학습은 두 가지 유형으로 나뉘는데, 분류와 회귀로 나뉨. 두 가지 기법의 차이는 분류는 카테고리와 같은 이산형, 회귀는 연속형 숫자값.

회귀모델:

1. 일반 선형 회귀: 예측값과 실제 값의 RSS최소화. 규제 적용X
2. 릿지: L2 규제를 추가함.
3. 라쏘:L1 규제 추가함
4. 엘라스틱넷: L2,L1 규제 함께 적용
5. 로지스틱 회귀: 분류에 활용되는 선형 모델이라 할 수 있음.
6.

2.단순 선형 회귀를 통한 회귀 이해

단순 선형 회귀: 독립 변수 1개, 종속 변수 1개.→ 1차 함수로 기울기와 절편이 회귀 계수.

실제 값과 회귀 모델의 차이에 따른 오류 값은 잔차라고함. 최적의 회귀 모델을 만드는 것은 잔차가 최소가 되는 모델을 만든다는 의미. → 잔차가 최소가 될 수 있는 회귀 계수를 찾는다는 의미.

오류를 더할 때는 절댓값을 취하거나, 오류 값의 제곱을 더해 RSS 로 함. 보통 RSS 로 진행.

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

(i는 1부터 학습 데이터의 총 건수 N까지)

3. 비용 최소화-경사 하강법

경사 하강법은 고차원 방정식에대한 문제를 해결해 주면서 비용 함수 RSS를 최소화하는 방법을 직관적으로 제공하는 뛰어난 방식입니다. 사실 경사 하강법은 '데이터를 기반으로 알고리즘이 스스로 학습한다'는 머신러닝의 개념을 가능하게 만들어준 핵심 기법의 하나입니다. 경사 하강법의 사전적 의미인 '점진적인 하강'이라는 뜻에서도 알 수 있듯이, '점진적으로' 반복적인 계산을 통해 w 파라미터 값을 업데이트하면서 오류 값이 최소가 되는 W 파라미터를 구하는 방식입니다.

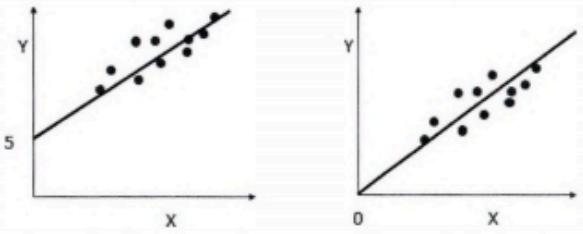
*경사하강법과 확률적 경사하강법: 예측 성능상의 큰 차이가 없음. 따라서 큰 데이터 처리 시, 경사 하강법은 매우 시간이 오래걸리므로 확률적 경사 하강법 사용.

*피처가 여러 개인 경우. → 피쳐가 1개인 경우를 확장해 도출함. 선형대수를 이용해 간단하게 예측값 도출 가능.

4. 사이킷런의 선형회귀를 이용한 보스턴 주택 가격 예측

사이킷런에는 LinearRegression클래스는 예측값과 RSS를 최소화해OLS 추정 방식으로 구현한 클래스.

```
class sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False, copy_X=True,
n_jobs=1)
```

입력 파라미터	**fit_intercept**: 불린 값으로, 디폴트는 True입니다. Intercept(절편) 값을 계산할 것인지 말지를 지정합니다. 만일 False로 지정하면 intercept가 사용되지 않고 0으로 지정됩니다. fit_intercept=True fit_intercept=False  **normalize**: 불린 값으로 디폴트는 False입니다. fit_intercept가 False인 경우에는 이 파라미터가 무시됩니다. 만일 True이면 회귀를 수행하기 전에 입력 데이터 세트를 정규화합니다.
속성	**coef_**: fit() 메서드를 수행했을 때 회귀 계수가 배열 형태로 저장하는 속성. Shape는 (Target 값 개수, 피처 개수). **intercept_**: intercept 값

*다중 공선성: 피처 간의 상관관계가 매우 높은 경우, 분산이 매우 커져서 오류에 매우 민감해지는 것을 말함. 따라서 이를 고려하여 중요한 피처만 남기고 제거하거나 규제를 적용하는 것이 필요함. PCA를 통해 차원 축소를 수행하는 것도 고려해 볼 수 있음.

회귀 평가 지표: 실제 값과 회귀 예측값의 차이 값을 기반으로 한 지표가 중심. 절댓값, 제곱, 제곱 후 루트를 활용. 혹은 MSE와 MSLE에 로그를 적용한 RMSLE도 있음.

평가 지표	설명	수식
MAE	Mean Absolute Error(MAE)이며 실제 값과 예측값의 차이를 절댓값으로 변환해 평균한 것입니다.	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert Yi - \hat{Y}i\rvert$
MSE	Mean Squared Error(MSE)이며 실제 값과 예측값의 차이를 제곱해 평균한 것입니다.	$MSE = \frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2$
RMSE	MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있으므로 MSE에 루트를 씌운 것이 RMSE(Root Mean Squared Error)입니다.	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}i)^2}$
R^2	분산 기반으로 예측 성능을 평가합니다. 실제 값의 분산 대비 예측값의 분산 비율을 지표로 하며, 1에 가까울수록 예측 정확도가 높습니다.	$R^2 = \frac{\text{예측값 } Variance}{\text{실제값 } Variance}$

사이킷런이 제공하는 지표:

평가 방법	사이킷런 평가 지표 API	Scoring 함수 적용 값
MAE	metrics.mean_absolute_error	'neg_mean_absolute_error'
MSE	metrics.mean_squared_error	'neg_mean_squared_error'
RMSE	metrics.mean_squared_error를 그대로 사용하되 squared 파라미터를 False로 설정.	'neg_root_mean_squared_error'
MSLE	metrics.mean_squared_log_error	'neg_mean_squared_log_error'
R^2	metrics.r2_score	'r2'

과거 버전의 사이킷런은 RMSE를 계산하는 함수가 제공되지 않았지만, 0.22 버전부터는 RMSE를 위한 함수를 제공합니다. RMSE를 구하기 위해서는 MSE를 위한 metrics.mean_squared_error( ) 함수를 그대로 사용하되, squared 파라미터를 False로 지정해 사용합니다. mean_squared_error( ) 함수는 squared 파라미터가 기본적으로 True입니다. 즉 MSE는 사이킷런에서 mean_squared_error(실제값, 예측값, squared=True)이며 RMSE는 mean_squared_error(실제값, 예측값, squared=False)를 이용해서 구합니다.

5. 다항 회귀와 과적합/과소적합의 이해

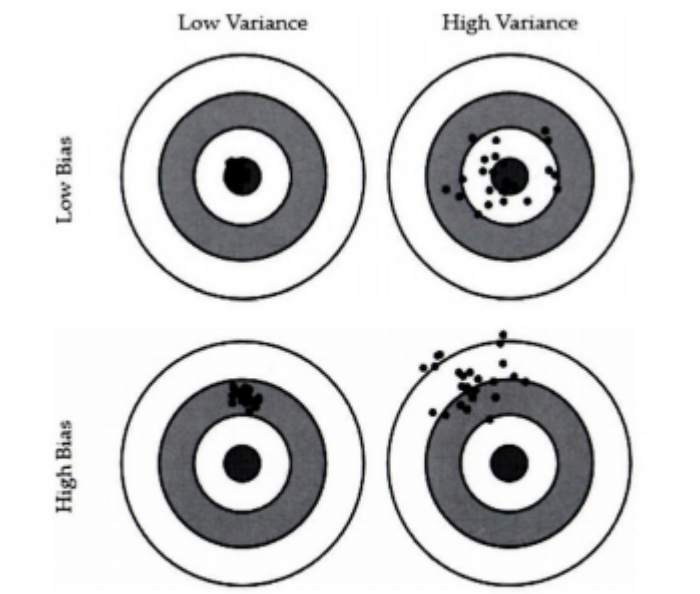
한 가지 주의할 것은 다항 회귀를 비선형 회귀로 혼동하기 쉽지만, 다항 회귀는 선형 회귀라는 점입니다. 회귀에서 선형 회귀/비선형 회귀를 나누는 기준은 회귀 계수가 선형/비선형인지에 따른 것이지 독립변수의 선형/비선형 여부와는 무관.

사이킷런은 다항회귀를 위한 클래스를 제공하지 않음. PolynomialFeatures 클래스를 통해 피처를 Polynomial(다항식) 피처로 변환.

다항 회귀의 차수(degree)를 높일수록 학습 데이터에만 너무 맞춘 학습이 이뤄져서 정작 테스트 데이터 환경에서는 오히려 예측정확도가 떨어집니다. 즉, 차수가 높아질수록 과적합의 문제가 크게 발생합니다.

편향-분산 트레이드 오프

저편향 고편향 저분산 고분산: 일방적으로 편향과 분산은 한쪽이 높으면 한쪽이 낮아지는 경향이 있음. 편향이 높으면 분산은낮아지고（과소적합） 반대로 분산이 높으면 편향이 낮아집니다（과적합）.



〈 편향과 분산의 고/저에 따른 표현.
http://scott.fortmann-roe.com/docs/BiasVariance.html에서 발췌 〉

6. 규제 선형 모델-릿지,라쏘,엘라스틱넷

회귀모델은 적절히 데이터에 적합하면서도 회귀 계수가 기하급수적으로 커지는 것을 제어할 수 있어야 함.RSS 최소화 방법과 과적합을 방지하기 위해 회귀 계수 값이 커지지 않도록 하는 방법이 서로 균형을 이루는 것이 중요함.

회귀 계수의 크기를 제어해 과적합을 개선하려면

alpha를 0에서부터 지속적으로 값을 증가시키면 회귀 계수 값의 크기를 감소시킬 수 있음.

규제는 크게 L2 방식과 L1 방식으로 구분됩니다. L2 규제 W의 제곱에 대해 페널티를 부여하는 방식을 말합니다. L2 규제를 적용한 회귀를 릿지(Ridge) 회귀라고 합니다.

라쏘(Lasso) 회귀는 L1 규제를 적용한회귀입니다. L1 규제는 W의 절댓값에 대해 페널티를 부여합니다. L1 규제를 적용하면 영향력이 크지 않은 회귀 계수 값을0으로 변환

엘라스틱넷 회귀 : L2 규제와 L1 규제를 결합한 회귀

ElasticNet 클래스를 통해서 엘라스틱넷 회귀를 구현.

선형 회귀 모델을 위한 데이터 변환

선형 회귀 모델의 기본 가정

- **선형 관계 가정:**

  피처(feature)와 타깃(target) 간의 관계가 **직선적(선형)**이라고 가정.

  → 최적의 선형함수를 찾아 예측 수행.

- **정규 분포 선호:**

  선형 회귀는 입력 데이터(피처)와 출력값(타깃)이 **정규 분포(Normal distribution)** 형태일수록 성능이 좋음.

- **왜곡된(Skewed) 분포 문제:**
  - 타깃값의 분포가 한쪽으로 치우치면(왜도 존재) → **예측 성능 저하**
  - 피처의 분포가 왜곡되어도(정도는 작지만) 예측력에 부정적 영향 가능

데이터 변환(스케일링/정규화)의 필요성

- 데이터 분포를 **정규 분포에 가깝게 변환**하면 모델 성능 향상 가능.
- 단, **항상 효과적인 것은 아님.**

  → 피처나 타깃의 분포가 **심하게 왜곡된 경우에만** 유용.

피처 데이터 변환 방법 (사이킷런 사용)

방법	설명	특징
① **StandardScaler**	평균 0, 분산 1의 표준 정규 분포로 변환	값의 중심을 0 근처로 조정
② **MinMaxScaler**	최솟값 0, 최댓값 1로 정규화	값의 범위를 일정 구간으로 제한
③ **PolynomialFeatures**	스케일링된 데이터에 다항식 피처 추가	비선형 관계 보강, 과적합 주의
④ **로그 변환 (Log Transformation)**	`log(x+1)` 적용으로 정규 분포에 근접	왜곡된 분포 교정에 **가장 많이 사용**

로그 변환(Log Transformation)의 중요성

- 피처뿐 아니라 **타깃값에도 자주 적용됨**
- 타깃이 왜곡된 분포일 때 로그 변환을 적용하면:
  - 분포가 완화되어 예측 안정성 증가
  - 회귀 모델의 **성능 향상 사례 다수 존재**

- 주의: 로그 변환된 타깃값을 다시 원래 값으로 되돌리는 것은 복잡할 수 있음 (`exp()` 필요)

구분	주요 목적	장점	주의점
StandardScaler	평균·분산 기준 정규화	수렴 안정	큰 향상 기대 어려움
MinMaxScaler	범위(0~1) 조정	단위 차이 제거	이상치(outlier)에 민감
PolynomialFeatures	비선형 관계 반영	예측력 향상 가능	피처 급증 → 과적합
Log Transform	왜곡된 분포 교정	**성능 향상 효과 큼**	0 이하 값은 변환 불가

7. 로지스틱 회귀

로지스틱 회귀는 선형 회귀 방식을 분류에 적용한 알고리즘입니다. 즉, 로지지스틱 회귀는 분류에 사용됨.

로지스틱 회귀 역시 선형 회귀 계열로 회귀가 선형인가 비선형인가는독립변수가 아닌 가중X](weight) 변수가 선형인지 아닌지를 따릅니다. 로지스틱 회귀가 선형 회귀와 다른 점은 학습을 통해선형 함수의 회귀 최적선을 찾는 것이 아니라 시그모이드(Sigmoid) 함수 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류를 결정한다는 것입니다.

→ 사이킷런의 LogistciRegression 클래스 활용.

→ solver 파라미터의 'lbfgs', 'liblineaf,'newton-eg', 'sag', 'saga' 값을 적용해서 최적화를 선택

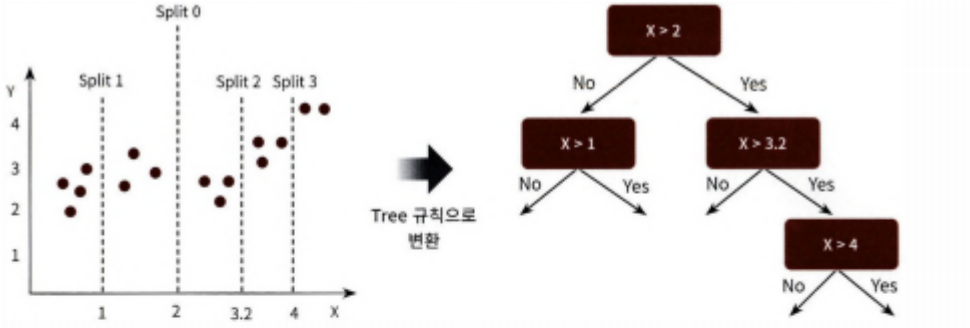
8. 회귀트리

회귀 함수를 기반으로 하지 않고 결정 트리와 같이 트리를 기반으로 하는 회귀 방식 존재.

원리: 규칙에 따라 분할 후, 평균값을 구해서 리프 노드에 결정 값으로 할당.

결정 트리, 랜덤 포레스트, GBM, XGBoost, LightGBM 등의 앞 4장의 분류에서 소개한 모든 트리 기반의 알고리즘은 분류뿐만 아니라 회귀도 가능

알고리즘	회귀 Estimator 클래스	분류 Estimator 클래스
Decision Tree	DecisionTreeRegressor	DecisionTreeClassifier
Gradient Boosting	GradientBoostingRegressor	GradientBoostingClassifier
XGBoost	XGBRegressor	XGBClassifier
LightGBM	LGBMRegressor	LGBMClassifier



노드 생성 기준에 부합하는 트리 분할이 완료됐다면 리프 노드에 소속된 데이터 값의 평균값을 구
최종적으로 리프 노드에 결정 값으로 할당합니다.