✅

Linear Regression

▼ 참고

이수안컴퓨터연구소 유튜브

회귀(Regression)예측

- 수치형 값을 예측(y값이 연속된 수치로 표현)

선형모델(Linear Models)

- 입력 데이터에 대한 선형 함수를 만들어 예측 수행

선형회귀(Linear Regression)

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)
```

- 최소제곱법(Ordinary Least Squares)라고도 한다
- 모델의 예측과 정답 사이의 **평균제곱오차(Mean Squared Error)를 최소화** 하는 학습 파라미터 w를 찾는다
- 평균제곱오차는 다음과 같이 정의

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

y: 정답
$\hat{y}$: 예측 값을 의미

다양한 오류 측정 방법

- ME(Mean Error)

$$ME = \frac{\sum_{i=1}^{n} e_i}{n} = \frac{\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)}{n}$$

  - 예측 오차의 평균 → 양수와 음수가 상쇄된다
- MSE(Mean Squared Error)

$$MSE = \frac{\sum (y - \hat{y})^2}{n}$$

  - 특이값에 민감하다 → 특이값이 존재하면 값이 커진다
- MAE(Mean Absolute Error)

$$MAE = \frac{\sum |y - \hat{y}|}{n}$$

  - 절대값을 취하기 때문에 가장 직관적이다
  - MSE보다 특이값에 robust하다
- RMSE(Root Mean Squared Error)

$$RMSE = \sqrt{\frac{\sum (y - \hat{y})^2}{n}}$$

  - MSE에 루트를 씌워서 실제값과 유사한 단위로 바꾼다
  - MSE보다 해석이 쉽다
- MAPE(Mean Absolute Percentage Error)

$$MAPE = \frac{\sum \left|\frac{y - \hat{y}}{y}\right|}{n} * 100\%$$

  - MSE보다 특이값에 robust하다

    - why percentage?
        - MAE|RMSE → 절대적 오차
        - percentage → 상대적 오차
- MPE(Mean Percentage Error)

$$MPE = \frac{\sum_{i=1}^{n} \frac{Y_i - \hat{Y}_i}{Y_i} \cdot 100}{n}$$

일반적인 모델 평가 방법

- 오류 측정 방법(eg. RMSE)
- 결정계수(R^2, coefficient of determination)

```
from sklearn.metrics import mean_squared_error, r2_score

predict = model.predict(X_train)
rmse = (np.sqrt(mean_squared_error(y_train, predict)))
r2 = r2_score(y_train, predict)

print('RMSE: {}'.format(rmse))
print('R2 Score: {}'.format(r2))
```

- 생성된 회귀 모델에 대해 평가를 위해 LinearRegression 객체에 포함된 두 개의 속성값을 통해 수식을 표현

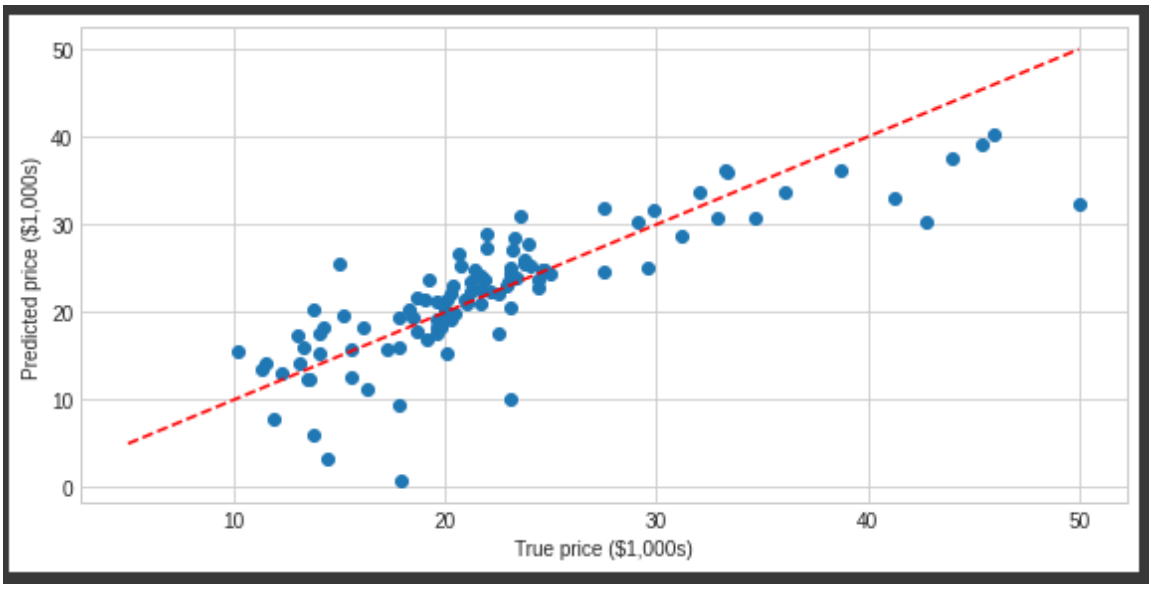
intercept_: 추정된 상수항
coef_: 추정된 가중치 벡터

- 시각화

```
def plot_boston_prices(expected, predicted):
    plt.figure(figsize=(8,4))
    plt.scatter(expected, predicted)
    plt.plot([5, 50], [5, 50], '--r') # describe로 최대'최소값 range 확인
    plt.xlabel('True price ($1,000s)')
    plt.ylabel('Predicted price ($1,000s)')
    plt.tight_layout()

predicted = model.predict(X_test)
expected = y_test
```

```
plot_boston_prices(expected, predicted)
```



교차검증(Cross Validation)

- 데이터의 양이 적고 분리가 잘 안된 경우에는 잘못된 검증이 될 수 있다
- 그래서 테스트셋을 여러개로 구성하여 교차 검증을 진행한다
- 모델 오류 측정 점수로 NMSE를 사용
  - followed by sklearn convention regarding scoring: `the higher the better`
    minimizing MSE is equivalent to maximizing negative-MSE

```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(model, boston.data, boston.target,
                         cv=10, scoring='neg_mean_squared_error')
print('NMSE scores: {}'.format(scores))
print('NMSe scores: mean: {}'.format(scores.mean))
print('NMSe scores: std: {}'.format(scores.std))
```

규제(Regularization)

- 학습이 과대적합 되는 것을 방지하고자 일종의 penalty를 부여하는 것
  가중치를 규제하면 모델의 일반화 성능이 올라간다
  훈련 데이터 점수는 낮을 수 있다
- L2규제가 L1규제에 비해 더 안정적이라 일반적으로는 L2규제가 더 많이 사용된다

릿지(Ridge) - L2 규제

$$Error = MSE + \alpha w^2$$

라쏘(Lasso) - L1 규제

$$Error = MSE + \alpha |w|$$

- L2규제(릿지)
  - 각 가중치 제곱의 합에 규제 강도(Regularization Strength)λ를 곱한다
  - λ를 크게 하면 가중치가 더 많이 감소(규제를 중요시함)
  - λ를 작게 하면 가중치가 증가(규제를 중요시하지 않음)
- L1규제(라쏘)
  - 각 가중치의 합에 규제 강도(Regularization Strength)λ를 곱하여 오차에 더한다
  - 어떤 가중치(w)는 실제로 0이 된다 → 모델에서 완전히 제외된다
    `feature selection`으로 사용할 수 있다

다양한 선형회귀모델

- 릿지|라쏘|엘라스틱넷|직교정합추구|다항회귀
  - 선형회귀를 개선한 선형모델
  - 선형회귀와 비슷하지만, `가중치의 절대값을 최대한 작게 만든다`는 것이 다르다
  - `각각의 특성(feature)이 출력값에 주는 영향을 최소한으로 만들도록 규제(regularization)`를 거는 것
  - 규제를 사용하면 `다중공선성(multicollinearity) 문제를 방지`하기 때문에 모델의 `과대적합을 막을 수 있게 된다`
  - 다중공선성 문제는 두 특성이 일치에 가까울 정도로 관련성(상관관계)이 높을 경우 발생

릿지회귀(Ridge Regression)

- 규제

$$RidgeMSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \alpha \sum_{i=1}^{P} w_i^2$$

α: 사용자가 지정하는 매개변수
α가 크면 규제의 효과가 커지고, α가 작으면 규제의 효과가 작아짐

라쏘회귀(Lasso Regression)

- 규제

$$LassoMSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \alpha\sum_{i=1}^{p}|w_i|$$

라쏘 회귀도 매개변수인 α 값을 통해 규제의 강도 조절 가능

엘라스틱넷(ElasticNet)

- 규제

$$ElasticMSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) + \alpha\rho\sum_{i=1}^{p}|w_i| + \alpha(1-\rho)\sum_{i=1}^{p}w_i^2$$

α: 규제의 강도를 조절하는 매개변수
ρ: 라쏘 규제와 릿지 규제 사이의 가중치를 조절하는 매개변수

- 릿지회귀와 라쏘회귀, 두 모델의 모든 규제를 사용하는 선형모델
- 규제항은 릿지와 회귀의 규제항을 단순히 더한 것으로 사용
  r=0 → 릿지회귀, r=1 → 라쏘회귀
- 데이터 특성이 많거나 서로 상관관계가 높은 특성이 존재할 때 위의 두 모델보다 좋은 성능을 보여준다

직교정합추구(Orthogonal Matching Pursuit)

- 규제방법 1

$$\arg\min_{w} ||y - \hat{y}||_2^2 \; subject\ to\ ||w||_0 \leq k$$

$||w||_0$: 가중치 벡터 w에서 0이 아닌 값의 개수

  - 가중치 벡터 w에서 0이 아닌 값이 k개 이하가 되도록 훈련

    `k개 이하를 제외한 나머지 가중치 벡터를 제거`

- 규제방법 2

$$\arg\min_{w} ||w||_0 \ subject\ to\ ||y-\hat{y}||_2^2 \leq tol$$

$||y-\hat{y}||_2^2$는 $\sum_{i=1}^{N}(y-\hat{y})^2$와 같은 의미

  - 오차 제곱합을 tol 이하로 하면서 $||w||_0$를 최소로 하는 모델

- 모델에 존재하는 가중치 벡터에 특별한 제약을 거는 방법
- `모델에 필요 없는 데이터 특성을 훈련 과정에서 자동으로 제거`하도록 만들 수 있다

다항회귀(Polynomial Regression)

- `입력 데이터를 비선형 변환 후 사용`하는 방법
- 모델 자체는 선형 모델

$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1^2 + w_5x_2^2$$

- 차수가 높아질수록 더 복잡한 데이터 학습 가능
- [a,b] 2개의 feature가 존재하고 degree=2라면 polynomial features는 $[1, a, b, a^2, ab, b^2]$이 된다

편차|오차|잔차

- 편차(deviation)
  - 평균 & 관측치의 차이
- 오차(error)
  - 모집단 회귀식 추청치 & 관측치의 차이
- 잔차(residual)

- 표본의 회귀식 추정치 & 관측치의 차이

다중회귀분석

- 독립변수가 두 개 이상인 회귀 분석
- 회귀모델에 포함되는 예측변수 선정 기준은 다음과 같다
  - 종속변수와 높은 상관관계
  - 독립변수 간 낮은 상관관계 → 다중공선성 문제를 회피해야 한다
  - 예측변수 개수는 적을수록 유리하다

다중공선성

- 독립변수 간 상관관계가 높은 상태
- 측정지표
  - 공차한계(Tolerance)

$$1 - R_i^2$$

  - VIF

$$\frac{1}{1 - R_i^2}$$

    - 공차한계의 역수
    - 일반적으로 공차한계 0.1 이하, VIF 10 이상 시 다중공선성을 보인다고 판단

상관계수|결정계수|수정결정계수

- 상관계수

$$\mathrm{R} = \frac{\mathrm{cov(x,y)}}{\sqrt{\mathrm{var(x)var(y)}}} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

  - -1~1 사이 값을 가진다

    - 두 변수 간 단순 관련성이 아닌 선형적인 관계를 나타낸다
      `상관계수 값이 0에 가까운 값을 가졌더라도 두 변수 간 비선형적인 관계가 있을 수 있다`

- SSR(Sum of Sqared Residual)

$$SSR = \sum_{i=1}^{n} \left(y_i - \hat{y_i}\right)^2$$

    - 잔차(residual)의 제곱합
    - `회귀선에 의해 설명되는 변동`을 의미한다

- SST(Sum of Squared Total)

$$SST = \sum_{i=1}^{n} \left(y_i - \overline{y}\right)^2$$

    - 종속변수의 분산 → 평균과의 차이를 제곱한 것의 합
    - `총변동`을 의미한다

- 결정계수(R^2, coefficient of determination)
    - 구하는 방법은 두 가지가 있다

$$r^2 = (r_{XY})^2$$

$$\mathrm{R}^2 = \frac{\sum(\hat{y_i} - \bar{y})}{\sum(y_i - \bar{y})^2} = \frac{\textit{회귀선에 의해 설명되는 변동}}{\textit{전체 변동}}$$

    - 회귀모델 적합도(goodness of fit)를 평가
      `회귀모델이 종속변수를 얼마나 설명하는가?`
    - 종속변수에 대해 설명변수의 설명력을 평가
    - r이 -1~1 사이 값을 가지므로 r^2은 0~1 사이 값을 가진다
    - 값이 0.65보다 크면 유용성이 높다고 할 수 있다
      → Y의 변동은 x의 변동에 의해 65%정도 설명된다고 해석

- 수정결정계수

$$\text{adjusted } R^2 = 1 - \frac{n-1}{(n-p-1)(1-R^2)}$$

  - 결정계수가 가진 문제를 보완한 계수
    - 결정계수는 독립변수 개수가 많아질수록 값이 커진다
      종속변수 변동을 잘 설명하지 못하는 변수가 모델에 추가되더라도 값이 커질 수 있다
  - 표본의 크기와 독립변수의 수를 고려하여 계산
  - 단순회귀분석의 경우 일반결정계수를 사용하면 되지만, 다중회귀분석의 경우 수정결정계수를 함께 고려하는 것이 좋다
  - 표본 크기가 200개 이상일 때는 일반결정계수와 차이가 미미하다