1.1 이론

1.1.0 머신러닝 파이프라인 개요

머신러닝 모델을 구축하고 적용하는 과정은 단순히 알고리즘을 선택하고 실행하는 것을 넘어, 다음과 같은 **전체 파이프라인 과정**으로 구성됩니다.

전체 흐름

> 데이터 수집 → 데이터 전처리 → 특성 선택 → 모델 선택 → 학습 → 평가 → (재)튜닝 → 예측/활용

각 단계 설명

1. 데이터 수집

- CSV, Excel, 데이터베이스, 센서 등 다양한 소스에서 데이터 수집
- 예: Kaggle의 부동산 데이터, 공공 데이터 포털

2. 데이터 전처리

- 결측치 처리, 이상치 제거, 범주형 인코딩, 정규화 등
- 예: `StandardScaler`, `OneHotEncoder`, `SimpleImputer`

3. 특성 선택 및 차원 축소

- 모델 성능에 영향을 주는 핵심 변수 선별
- PCA 같은 차원 축소 기법 사용

4. 모델 선택 및 학습

- 선형 회귀, 의사결정트리, 로지스틱 회귀 등 문제에 맞는 모델 선택
- 훈련 데이터를 통해 모델 학습

5. 성능 평가

- 테스트셋을 사용하여 예측 성능 평가
- 회귀: MSE, MAE, R^2 / 분류: 정확도, F1, ROC-AUC

6. 모델 튜닝

- 하이퍼파라미터 조정, 특성 재선택 등으로 성능 향상
- `GridSearchCV`, `RandomizedSearchCV` 활용

7. 예측 및 활용

- 실제 문제에 적용 (미래 값 예측, 분류 자동화 등)

1.1.1 회귀 분석이란?

회귀 분석은 주어진 특성(입력 변수)을 바탕으로 **수치적인 연속값을 예측**하는 지도학습(Supervised Learning)의 대표적인 방법입니다.

예시:

- 집의 면적과 방 수로 가격 예측
- 수면 시간과 학습 시간으로 수능 점수 예측
- 환경 데이터로 탄소 배출량 예측

지도학습의 두 형태 정리:

구분	목표값	예시	대표 알고리즘
회귀	연속값 (수치형)	가격, 점수, 시간	선형 회귀, 다항 회귀 등
분류	범주 (class label)	합/불합, 고양이/개	로지스틱 회귀, SVM 등

1.1.2 선형 회귀 모델의 수식 구조

선형 회귀(Linear Regression)는 가장 기본적인 회귀 모델로, 입력 변수들과 출력 변수 사이에 **선형 관계**가 있다고 가정합니다.

가설 함수 수식:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

여기서 θ_j 는 모델이 학습해야 할 계수이며, x_j 각 특성(feature)입니다.

1.1.3 비용 함수 (Loss Function)

모델이 얼마나 잘 예측했는지를 측정하기 위해 가장 대표적인 손실 함수로 평균 제곱 오차(MSE)를 사용합니다.

비용 함수 (MSE):

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

m: 전체 학습 데이터의 개수

$x^{(i)}$: i 번째 입력 샘플

$y^{(i)}$: i 번째 실제 정답값

$h_\theta(x^{(i)})$: i 번째 예측값

$J(\theta)$: 파라미터 θ에 대한 비용 함수

1.1.4 경사 하강법 (Gradient Descent)

오차를 줄이기 위해 θ를 반복적으로 업데이트합니다. 미분을 통해 오차의 기울기를 계산하고 그 반대 방향으로 파라미터를 반복적으로 갱신합니다.

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

θ_j: j번째 파라미터 (가중치)

α: 학습률 (learning rate)

$\frac{\partial J}{\partial \theta_j}$: θ_j에 대한 비용 함수의 미분값

1.1.5 다항 회귀 (Polynomial Regression)

입력 x를 다항식으로 확장해 더 복잡한 비선형 관계를 표현할 수 있습니다. 선형회귀는 입력과 출력 간 관계가 직선일때만 잘 작동합니다. 다항식으로 확장이 되면 곡선 형태의 데이터에 대해서도 잘 작동할 수 있습니다.

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \cdots + \theta_d x^d$$

x: 입력 변수

d: 다항식 차수 (degree)

θ_i: 다항 항(i차 항)의 계수

1.1.6 정규화 회귀 (Regularized Regression)

다항 회귀나 고차원 데이터에서는 모델이 훈련 데이터에만 너무 잘 맞는 **과적합(overfitting)** 문제가 자주 발생합니다. 이를 해결하기 위해 비용 함수에 **패널티 항**을 추가하는 방식이 정규화입니다.

(1) Ridge 회귀 (L2 정규화)

> ridge : 산등성이, 산마루

오차를 최소화하면서 동시에 파라미터들이 제곱합이 너무 커지지 않도록 제한합니다.

$$J_{ridge}(\theta) = J(\theta) + \lambda \sum_{j=1}^{n} \theta_j^2$$

λ: 정규화 강도 (규제 계수) : 값이 클 수록 제약이 강해짐

$\sum_{j=1}^{n} \theta_j^2$: 파라미터 제곱합 (L2 norm) : 파라미터를 0에 가깝게 만들지만, 완전히 0으로 만들진 않음.

(2) Lasso 회귀 (L1 정규화)

> LASSO : Least Absolute Shrinkage and Selection Operator

오차를 최소화하면서 파라미터 절댓값의 합이 너무 커지지 않도록 제한합니다.

$$J_{lasso}(\theta) = J(\theta) + \lambda \sum_{j=1}^{n} |\theta_j|$$

$\sum_{j=1}^{n} |\theta_j|$: 파라미터 절댓값의 합 (L1 norm)

일부 파라미터가 완전히 0이 되기 때문에, 불필요한 특성을 제거하는 효과가 있음.

고차원에서의 변수 선택 역할 수행 가능

1.1.7 회귀 모델 비교 요약

모델	과적합 제어	특성 선택	해석 용이성	표현력
선형 회귀	×	×	높음	낮음
다항 회귀	×	×	중간	높음
릿지 회귀	O (L2)	×	중간	중간
라쏘 회귀	O (L1)	O	중간	중간

- **과적합 제어**: 모델이 훈련 데이터에만 과도하게 맞춰지는 현상(과적합)을 방지하기 위해 정규화 항(릿지:L2, 라쏘:L1)을 사용하여 파라미터 크기를 제한하는 방법입니다.
- **특성 선택**: 예측에 중요하지 않은 특성(feature)의 가중치를 0으로 만들어 유용한 특성만 자동으로 선택하는 방법이며 라쏘(L1) 회귀에서 수행됩니다.
- **해석 용이성**: 모델이 어떤 특성이 결과에 얼마나 영향을 미쳤는지 명확한 수치(계수 등)로 제시하여 사용자가 예측 과정과 이유를 직관적으로 이해할 수 있는 정도를 말합니다.
- **표현력**: 데이터가 가진 복잡한 패턴과 비선형적 관계를 모델이 얼마나 유연하고 정확하게 표현할 수 있는지를 나타내는 모델의 능력을 의미합니다.