AI기반 데이터 분석 및 AI Agent 개발 과정

『1과목 : 』
AI기반 데이터 분석

2025.09.22-10.02(9일, 62시간)

Prepared by DaeKyeong K

Ph.D.

Copyright 2022. DaeKyeong all rights reserved

삼성전자 구미사업부 G.제조팀

HS Global
HARDWARE SOFTWARE

목차

『1-6』
데이터 준비(Data Preparation)

Data Integration (통합)

Data Reduction (축소)

Data Transformation (변환)

Feature Engineering & Data Encoding

Cross Validation & Data Splitting

Data Quality Assessment & Model Performance Evaluation

Copyright 2022. DaeKyeong all rights reserved

개요

학습목표

- 이 워크샵에서는 데이터 통합(integration)에 대해 알 수 있습니다.

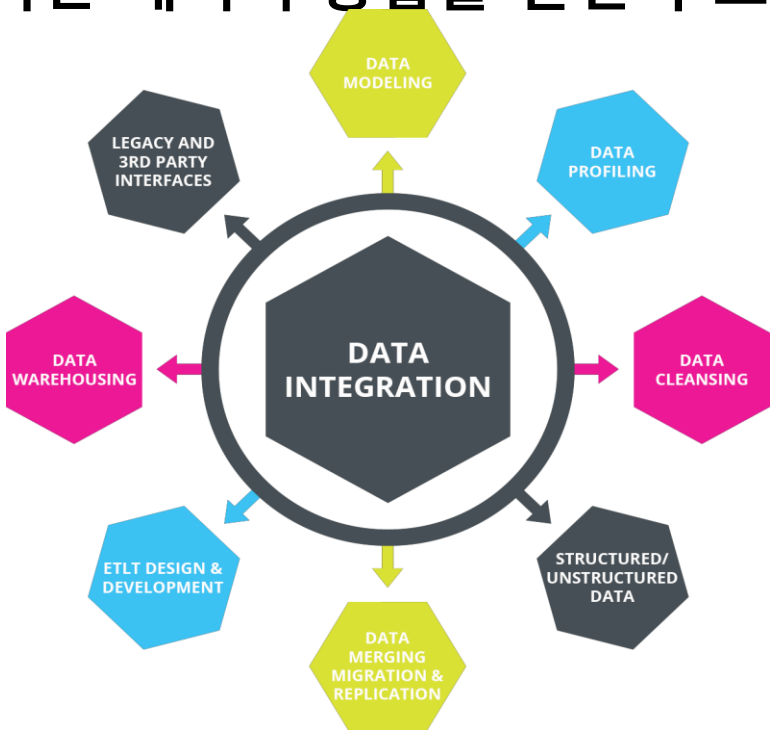
눈높이 체크

- 데이터 통합(integration)에 대해 들어보셨나요?

1. Data Integration?

데이터 통합Data Integration

- 서로 다른 출처의 여러 데이터를 결합
- 서로 다른 데이터 세트가 호환이 가능하도록 통합
- 같은 객체, 같은 단위나 좌표로 데이터를 통합
- 링크드 데이터의 핵심 목표 중 하나는 데이터 통합을 완전히 또는 거의 완전히 자동화하는 것
  - 링크드 데이터는 "웹 기술(URI, HTTP, RDF 등)"을 사용해 기계가 이해하고 탐색할 수 있도록 상호 연결된 데이터



출처 : http://www.matricis.com/en/integration-solutions/data-integration/

1. Data Integration?

데이터 통합Data Integration

- 데이터 통합data integration은 여러 데이터 저장소로부터 온 데이터의 합병
  - 데이터 웨어하우스data warehouse나 데이터마이닝data mining 같은 데이터 분석 작업은 다수의 원천 데이터로부터 하나의 통일된 데이터 저장소로 결합시키는 통합 작업 필요
  - 데이터 원천은 데이터베이스, 데이터 큐브data cube, 플랫 파일flat file 등 다양한 형태로 존재
  - 여러 데이터 원천들로부터 데이터를 통합할 때, 동일한 의미의 개체들이 서로 다르게 표현되어 있을 경우, 어떻게 일치 시킬 수 있을까? → 개체 식별 문제Entity Identification Problem

출처 : 개체 식별 문제Entity Identification Problem 이현호, Python과 SQL을 활용한 실전 데이터 전처리, 카오스북, 2018.

1. Data Integration?

데이터 값 충돌 탐지 및 해결

- 서로 다른 데이터 원천의 데이터들을 통합 할 때 동일한 개체에 대해서도 속성 값이 다를 수 있음 → 표현representation, 척도scaling, 부호화encoding 등의 차이
  - 거리를 나타내는 속성으로 어느 DB에서는 미터meter 단위로 다른 DB에서는 마일mile 단위로 저장
  - 학생 성적 데이터로 어느 DB에서는 과목별 점수가 저장되고, 다른 DB에서는 총점과 평균만 저장
- 동일한 개체의 동일한 값이 데이터 원천에 따라 다르게 표현되어 있는 경우, 데이터 통합 시에 기준을 정하여 데이터 값을 변환하여 통합시키는 것이 필요
- 통합 과정에서 DB의 속성을 일치시킬 때 데이터 구조에도 주의를 기울여야 함
- 원천 시스템의 기능적 종속성과 제약 사항들이 목표 통합 시스템의 것과 일치해야 함
  - 어느 시스템에서는 할인이 총 주문 금액에 적용되는 반면 다른 시스템에서는 주문을 구성하는 개별 항목에 적용
- 데이터의 의미적 이질성과 데이터 구조는 데이터 통합 시에 여러 문제를 발생시킴
- 여러 유형의 문제들을 신중하게 해결하여 통합 데이터의 중복과 불일치 문제를 최소화

1. Data Integration?

CSV

- CSV(쉼표로 구분된 값) 파일 형식은 데이터를 저장하고 공유하는 매우 간단한 방법.
  - CSV 파일은 데이터 테이블을 일반 텍스트로 보유.
  - 표(또는 스프레드시트)의 각 셀은 숫자나 문자열일 뿐. Excel 파일에 비해 CSV 파일의 주요 이점 중 하나는 일반 텍스트 파일을 저장, 전송 및 처리할 수 있는 프로그램이 많다는 것임. CSV 파일로 작업할 때 Excel의 일부 기능을 잃게 된다. Excel 스프레드시트의 모든 셀에는 정의된 "유형"(숫자, 텍스트, 통화, 날짜 등)이 있는 반면 CSV 파일의 셀은 원시 데이터일 뿐이다.

1. Data Integration?

CSV

- Python의 pandas library의 read_csv() 함수를 사용해서 외부 text 파일, csv 파일을 불러와서 DataFrame으로 저장. 불러오려는 text, csv 파일의 encoding 설정과 Python encoding 설정이 서로 맞지 않으면 UnicodeDecodeError 가 발생. 한글은 보통 'utf-8' 을 많이 사용하는데요, 만약 아래처럼 'utf-8' 코덱을 decode 할 수 없다고 에러 메시지가 나오는 경우가 있다.

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xc1 in position 26: invalid start byte

- w_list.to_csv('Test.csv',encoding='euc-kr') #한글 저장 잘 됨
- w_list.to_csv('Test.csv',encoding='utf-8') #한글 깨짐
- w_list.to_csv('Test.csv',encoding='utf-8-sig') #utf-8로 한글 저장 잘 됨

2. Data Correction(데이터 수정)

Data Correction(데이터 수정)

- Data Correction은 잘못 기록되었거나 불완전한 데이터를 정확하고 일관된 값으로 정정하는 작업
  - 왜 필요한가?

문제 유형	예시
오타	Jonuary → January
잘못된 숫자	나이: 250세, 수량: -5개
형식 오류	날짜 형식: 13/45/2023
코드 값 불일치	성별: M, male, 남 → 통일 필요
단위 혼용	170cm, 1.7m 등
중복 값	동일 인물 여러 행 존재

2. Data Correction(데이터 수정)

Data Correction(데이터 수정)

- Data Correction은 잘못 기록되었거나 불완전한 데이터를 정확하고 일관된 값으로 정정하는 작업
  - 왜 필요한가?

문제 유형	예시
오타	Jonuary → January
잘못된 숫자	나이: 250세, 수량: -5개
형식 오류	날짜 형식: 13/45/2023
코드 값 불일치	성별: M, male, 남 → 통일 필요
단위 혼용	170cm, 1.7m 등
중복 값	동일 인물 여러 행 존재

2. Data Correction(데이터 수정)

Data Correction(데이터 수정)

- Data Correction의 주요 작업

작업	설명	예시
오타 수정	문자열 유사도, 사전 기반 자동 교정	Jonh → John
값 정규화	표준값으로 통일	M, Male, 남자 → 남
범위 확인 및 수정	값이 비정상적이면 대체	나이 300 → 평균값으로 대체
형식 변환	일관된 날짜/시간/숫자 형식	2024/05/16 → 2024-05-16
결측값 보완	NULL → 평균값/이전값/예측값 등으로 채우기	NaN → 중앙값

2. Data Correction(데이터 수정)

Data Correction vs Data Cleaning

- Data Correction은 Cleaning의 하위 작업 중 하나이지만,

구분	설명
Data Cleaning	오류 제거, 중복 제거, 결측치 처리 전체 과정
Data Correction	잘못된 값을 "수정"하는 데 초점

『1-6』
데이터 준비(Data Preparation)

Data Integration (통합)

Data Reduction (축소)

Data Transformation (변환)

Feature Engineering & Data Encoding

Cross Validation & Data Splitting

Data Quality Assessment & Model Performance Evaluation

Copyright 2022. DaeKyeong all rights reserved

개요

학습목표

- 이 워크샵에서는 데이터 축소(Data Reduction에 대해 알 수 있습니다.

눈높이 체크

- 데이터 축소(Data Reduction) 에 대해 들어보셨나요?

1. Data Reduction?

데이터 축소Data Reduction

- 일반적으로 데이터는 매우 크기 때문에 대용량 데이터에 대한 복잡한 데이터 분석은 실 행하기 어렵거나 불가능한 경우가 많음
- 데이터 축소는 원래 용량 기준보다 작은 양의 데이터 표현결과를 얻게 되더라도 원 데이 터의 완결성을 유지하기 위해 사용
- 데이터를 축소하면 데이터 분석 시 좀 더 효과적이고 원래 데이터와 거의 동일한 분석 결 과를 얻어낼 수 있는 장점

출처 : https://www.cohesity.com/blog/cohesity-data-reduction-lock-stock-barrel/

1. Data Reduction?

데이터 축소Data Reduction

- 원천 데이터의 축소판(압축판)을 얻기 위한 데이터 부호화 또는 데이터 변환의 적용
  - 원천 데이터가 정보의 손실 없이 압축된다면 무손실 loseless
  - 원천 데이터의 근사치만으로 축소된다면 손실 lossy
  - 일반적으로 많이 사용되며 효과적인 손실 차원 축소 방법
    - 웨이블릿 변환 wavelet transform
    - 주성분 분석 principal components analysis

출처 : 이현호, Python과 SQL을 활용한 실전 데이터 전처리, 카오스북, 2018.

2. Data Reduction 종류

웨이블릿 변환wavelet transform

- 이산 웨이블릿 변환discrete wavelet transform, DWT: 데이터 벡터 X를 다른 수치적 벡터numerically vector X'으로 변환 (X와 X' 의 길이는 동일)
  - 각 튜플을 n 차원 데이터 벡터로 간주하면, 벡터 $X = x1, x2, \cdots, xn$를 각 튜플로 고려, 웨이블릿 변환 데이터가 원천 데이터와 같은 길이(속성 수)를 가지지만 데이터 축소로 볼 수 있는 것은 변환 데이터가 압축되어 보이기 때문
  - 웨이블릿 계수 중 가장 유력한 일부만을 저장함으로써 데이터 근사치를 유지
  - 예를 들어, 사용자가 정한 어떤 임계값보다 큰 모든 웨이블릿 계수들만 값을 유지하고 나 머지 계수들을 0으로 간주하면, 결과적인 데이터 표현은 매우 희소해지며, 데이터 희소 성data sparsity은 데이터 연산의 복잡도를 크게 감소시킬 수 있음
  - 데이터의 주요 특징들은 보존하면서도 잡음을 제거하는 역할을 하기도 하므로 데이터 정제를 위해서도 효과적임

2. Data Reduction 종류

주성분 분석Principal Components Analysis, PCA

- 주성분 분석Principal Components Analysis, PCA은 n개의 속성을 가진 튜플(n 차원의 데이터 벡터)에 대하여 데이터를 표현하는데 최 적으로 사용될 수 있는 n 차원 직교벡터orthogonal vector들에 대한 k를 찾음 ($k \leq n$)→ 감소된 차원의 공간을 갖는 데이터 공간 생성 (차원 축소)

2. Data Reduction 종류

수량 축소 방법 -표본 추출sampling

- 큰 데이터 집합을 많은 수의 임의 데이터 샘플(부분집합)로 표현 가능
- 대용량 데이터 집합 D가 N개의 튜플을 포함하고 있다고 가정

- 비복원 단순 무작위표본Simple Random Sample WithOut Replacement, SRSWOR: D로부터 N개의 튜플 중 에서 임의의 s개를 취하는 방법으로서 모든 튜플들의 표본으로 추출될 확률은 같음
- 복원 단순 무작위표본Simple Random Sample With Replacement With Replacement, SRSWR: 각 튜플이 D로부터 추출될 때마다 기록된 후 다시 제자리로 복원replace 된다는 것을 제외하면 SRSWOR와 유사, 각 튜플은 추출된 다음에 다시 추출될 수 있도록 D에 되돌려짐

- 집략표본Cluster Sample: D에 있는 튜플들이 M개의 상호 배반적 군집 cluster으로 묶여 있는 가운데 s개의 군집을 단순 무작위로 추출 ($s < M$)
- 층화표본Stratified Sample: D가 층strata이라 불리는 상호 배반적 부분들로 분할되어 있다면, 각 층에서 하나씩 단순 무작위로 추출 (예, 고객의 나이 그룹 각각에 대하여 하나의 층이 생 성되어 있는 고객 데이터로부터 층화표본을 얻음)

2. Data Reduction 종류

수량 축소 방법-히스토그램histogram

- 구간화를 사용하여 데이터 분포의 근사치를 구하는 데이터 축소의 전형적 형태
- 속성 A의 데이터를 버킷bucket 혹은 빈bin이라 불리는 분리 집합disjoint subset으로 나눔
- 각 버킷이 단일한 속성 값/빈도의 쌍으로 표현되기도 하고, 주어진 속성에 대한 연속 범 위continuous range를 나타내기도 함
- 히스토그램은 희소 데이터나 밀집 데이터 모두에 효과적, 비대칭적 데이터와 균일한 데이터 모두 매우 효과적
- 단일 속성에 대한 히스토그램은 다중 속성에 대한 것으로 확장 가능
- 다차원 히스토그램에서는 속성 간의 의존성 포착 가능
- 일반적으로 5개 까지의속성을 가진 데이터의 근사치를 구하는 데 효과적이라고 알려짐

2. Data Reduction 종류

수량 축소 방법-군집화clustering

- 군집cluster이라는 그룹으로 나눔
  - 한 군집 내 객체들과는 유사하면서도 다른 군
  - 집 내 객체들과는 유사하지 않도록 군집화
  - 유사성은 공간 내에서 객체들이 어떻게 가까 운지의 관점에 따라 거리 함수에 기반하여 정 의
  - 클러스터의 품질은 지름diameter의 표현으로 나타내고, 지름은 클러스터의 두 객체 간 최대 거리로 표현
  - 클러스터 간 중심 거리centroid distance는 클러스터 중심 간 거리로서 클러스터 품질로 대체 측정
  - 클러스터의 지름은 짧을수록(클러스터 내 객 체 간의 유사성이 강할수록), 클러스터 간 중 심 거리는 길수록(클러스터 간 유사성은 약할 수록) 군집화의 품질이 높다고 볼 수 있음

2. Data Reduction 종류

주요 기법

- 데이터 축소 기법은 고차원 데이터의 복잡성을 줄여 연산 효율성을 높이고, 노이즈를 줄이며, 모델 성능을 향상시키는 데 사용. 적절한 데이터 축소 기법을 선택하여 데이터의 유용한 정보를 보존하면서 불필요한 정보를 제거하여 더 효율적인 분석을 수행할 수 있.

유형	기법	설명
속성 축소	Feature Selection	불필요한 컬럼 제거 (예: 상관관계 낮은 변수 제거)
	Feature Extraction	기존 속성들을 조합해 새로운 축소된 속성 생성 (예: PCA)
차원 축소	PCA (주성분 분석)	고차원 데이터를 저차원 공간으로 투영
	LDA, t-SNE, UMAP 등	클래스 분리 / 시각화 목적
표본 축소	샘플링 (Sampling)	전체 데이터에서 일부만 추출
중복 제거	Deduplication	동일하거나 유사한 레코드 제거
불필요 데이터 제거	Null, Noise 제거	의미 없는 값, 잡음 제거

2. Data Reduction 종류

특성 선택 (Feature Selection)

- 필터링 방법 (Filter Methods): 통계적 기준 또는 상관 관계를 사용하여 특성을 선택. 예를 들어, 분산 기준으로 특성을 선택하거나 목표 변수와의 상관 관계가 높은 특성을 선택.
- 래퍼 방법 (Wrapper Methods): 모델을 사용하여 특성의 부분 집합을 평가하는 방식으로 선택. 예를 들어, 순차적으로 특성을 추가하거나 제거하여 가장 좋은 결과를 얻는 특성을 선택.
- 임베디드 방법 (Embedded Methods): 모델 훈련 과정에서 특성 선택을 수행. 일부 머신러닝 알고리즘은 특성 선택을 위한 내부 메커니즘을 제공.

3. 차원 축소

차원 축소(Dimensionality Reduction)?

- 차원축소 기법은 대량의 빅데이터를 분석하는 데 있어, 분석대상이 되는 여러 변수들의 주요 정보는 최대한 유지하면서 데이터 세트의 변수의 개수를 줄이는 일련의 탐색적 분석 기법을 말한다.

  - 차원축소 기법은 하나의 완결된 분석 기법으로 사용되기보다는 다른 분석과정을 위한 전 단계나 분석 수행 후 결과를 개선하기 위한 방법 혹은 효과적인 시각화 등의 목적으로 사용되는 경우가 많다.
  - 데이터 차원(변수 세트)이 많은 상황에서 데이터에 대한 효과적인 시각화를 통한 통찰을 얻고자 할 때, 저차원(일반적으로 2차원)으로 투영시키거나 다른 지도학습 및 자율학습 등을 수행하는 과정에서 매우 많은 변수가 있는 상황, 즉 차원의 저주(The curse of Dimensionality)를 해결하기 위해 서로 상관성이 높거나 유사한 변수들을 공통 변수로 결합하여 분석을 수행하고자 할 때 주로 사용되는 경우가 많다.

3. 차원 축소

데이터 축소(Data Reduction)?

- 차원의 저주(The curse of dimensionality)란 고차원 공간에서 데이터 분석 및 모델링을 수행할 때 발생하는 여러 문제들을 가리키는 용어. 고차원 데이터에서는 몇 가지 문제가 발생할 수 있다.

1. 데이터 희소성(Sparsity of data): 고차원 공간에서는 데이터가 채워져 있는 공간보다 희소한 경향이 있습니다. 데이터 포인트 간의 거리가 멀어져서 데이터가 더 분산되고 샘플 간의 관계를 파악하기 어려워집니다.
2. 계산 복잡성(Computational complexity): 고차원 데이터는 고차원의 많은 특성(feature)을 가지므로, 모델 학습 및 예측에 필요한 계산 복잡성이 증가합니다. 이는 메모리 사용량과 연산 속도에 영향을 미치며, 학습 시간이 증가할 수 있습니다.
3. 과적합(Overfitting): 고차원 데이터에서는 모델이 학습 데이터에 너무 맞추어져 새로운 데이터에 대한 일반화 성능이 떨어질 수 있습니다. 모델이 불필요한 특성이나 잡음까지 학습하여 일반화 능력이 감소할 수 있습니다.
4. 차원의 저하(Dimensionality reduction): 고차원 데이터에서는 시각화와 같은 목적으로 데이터를 이해하기 어렵기 때문에, 차원 축소 기법이 필요할 수 있습니다. 이러한 기법은 주성분 분석(PCA), t-SNE 등을 포함합니다.

- 차원의 저주를 극복하기 위해서는 데이터 특성을 선택하거나 추출하여 차원을 줄이거나, 더 많은 데이터를 수집하여 희소성을 감소시키는 방법 등이 있다. 또한, 적절한 특성 선택, 차원 축소 기법, 모델의 복잡성 관리 등을 통해 차원의 저주에 대처할 수 있다.

3. 차원 축소

데이터 축소(Data Reduction)?

- 특성 선택(Feature Selection) 은 넓은 의미에서 **차원 축소 (Dimensionality Reduction) 의 한 예

구분	방식	설명	대표 기법
1. 특성 선택 (Feature Selection)	있는 변수 중 "고르기"	기존 변수 중 중요한 것만 남김	상관관계 기반, 분산 기준, SelectKBest, Lasso 등
2. 특성 추출 (Feature Extraction)	새로운 변수 " 만들기"	기존 변수들을 조합하여 새로운 특성을 생성	PCA, LDA, t-SNE, UMAP, AutoEncoder 등

3. 차원 축소

주성분 분석(PCA)

- 특성 추출과 차원 축소- 주성분 분석은 n개의 속성을 가진튜플(n 차원의 데이터 벡터)에 대하여 데이터를 표현하는데 최적으로 사용될 수 있는 n차원 직교벡터orthogonal vector들에 대한 k를 찾음($k \leq n$)→ 감소된 차원의 공간을 갖는 데이터 공간 생성(차원 축소)
- 주성분 분석 모형

$$Z_1 = \gamma_{11}X_1 + \gamma_{12}X_2 + \ldots\ldots\ldots + \gamma_{1p}X_p = \gamma_1^T X$$
$$Z_2 = \gamma_{21}X_1 + \gamma_{22}X_2 + \ldots\ldots\ldots + \gamma_{2p}X_p = \gamma_2^T X$$
$$\ldots\ldots \qquad \ldots\ldots\ldots \qquad \ldots\ldots$$
$$Z_q = \gamma_{q1}X_1 + \gamma_{q2}X_2 + \ldots\ldots\ldots + \gamma_{qp}X_p = \gamma_q^T X$$

$$\mathrm{max}imize_{\gamma 11,\ldots\ldots,\gamma 1p}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\gamma_{1j}x_{ij}\right)^2\right\}$$

$$sbj\ \sum_{j=1}^{p}\gamma^{2_{1j}} = 1$$

새로운 주성분으로 해당 주성분 분산을 최대화하는 선형조합임

3. 차원 축소

주성분 분석(PCA)

- 주성분 분석 예

ID	국어	수학
1	95	95
2	90	95
3	80	75
4	60	70
5	40	35
6	80	80
7	95	90
8	30	25
9	15	10
10	60	70
평균	64.50	64.50
분산	808	925

$$\frac{95 - 64.50}{28.425}$$

$$\frac{30.50}{28.425} \approx 1.075$$

$$Z = \frac{X - \mu}{\sigma}$$

ID	국어	수학
1	1.1	1.0
2	0.9	1.0
3	0.5	0.3
4	-0.2	0.2
5	-0.9	-1.0
6	0.5	0.5
7	1.1	0.8
8	-1.2	-1.3
9	-1.7	-1.8
10	-0.2	0.2
평균	0.0	0.0
분산	1	1

3. 차원 축소

주성분 분석(PCA)

- 주성분 분석 예

ID	국어	수학
1	1.1	1.0
2	0.9	1.0
3	0.5	0.3
4	-0.2	0.2
5	-0.9	-1.0
6	0.5	0.5
7	1.1	0.8
8	-1.2	-1.3
9	-1.7	-1.8
10	-0.2	0.2
평균	0.0	0.0
분산	1	1
공분산	0.98	0.98

분산-공분산 매트릭스
20개의 숫자를 단 4개로 만들어줌

$$\sum = \begin{matrix} \\ 국어 \\ 수학 \end{matrix} \begin{bmatrix} 국어 & 수학 \\ 1 & 0.98 \\ 0.98 & 1 \end{bmatrix}$$

고유값

$$\sum = det \begin{bmatrix} 1-\lambda & 0.98 \\ 0.98 & 1-\lambda \end{bmatrix}$$

$$(1-\lambda)^2 - 0.98^2 = 0.02$$

3. 차원 축소

주성분 분석(PCA)

- **주성분 분석 예**

고유값 구하기

즉,

$$\begin{vmatrix} 1-\lambda & 0.98 \\ 0.98 & 1-\lambda \end{vmatrix} = 0$$

행렬식 전개:

$$(1-\lambda)(1-\lambda) - (0.98)^2 = 0$$
$$(1-\lambda)^2 - 0.9604 = 0$$
$$(1-\lambda)^2 = 0.9604$$
$$1-\lambda = \pm 0.98$$

λ 값 계산

- $1-\lambda = 0.98 \quad \Rightarrow \quad \lambda = 0.02$
- $1-\lambda = -0.98 \quad \Rightarrow \quad \lambda = 1.98$

분산-공분산 행렬의 **고유값**은:

$$\lambda_1 = 1.98, \lambda_2 = 0.02$$

즉, 파일에 나온 **0.02 값**은 이 고유값 계산에서 나온 **작은 고유값**.

『1-6』
데이터 준비(Data Preparation)

Data Integration (통합)

Data Reduction (축소)

Data Transformation (변환)

Feature Engineering & Data Encoding

Cross Validation & Data Splitting

Data Quality Assessment & Model Performance Evaluation

Copyright 2022. DaeKyeong all rights reserved

개요

학습목표

- 이 워크샵에서는 데이터 변환(Data Transformation)에 대해 알 수 있습니다.

눈높이 체크

- 데이터 변환(Data Transformation) 에 대해 들어보셨나요?

1. 데이터 변환(Data Transformation)

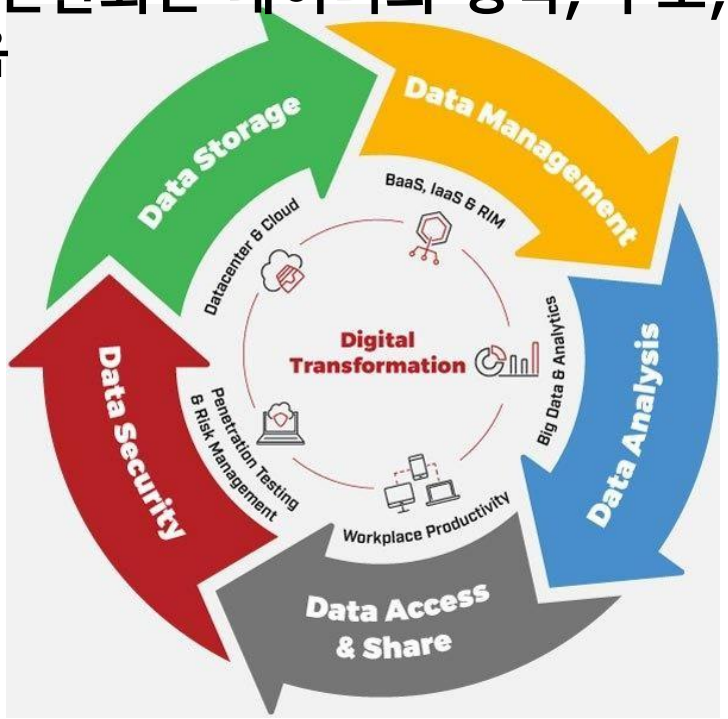
데이터 변환(Data Transformation)?

- 데이터 변환(Data Transformation)이란, 데이터를 변형하여 모델 학습에 적합한 형태로 만드는 과정.
  - 여기에는 주로 정규화(Normalization)와 표준화(Standardization)가 포함 됨.
  - 이러한 스케일링 기법들은 데이터의 스케일을 맞추고, 모델의 수렴을 빠르게 하거나 이상치에 덜 민감하게 만드는 등의 이점을 제공. 선택하는 스케일링 기법은 데이터와 모델에 따라 다르며, 주어진 문제에 가장 적합한 방법을 선택해야 함.

1. 데이터 변환(Data Transformation)

데이터 변환Data Transformation

- 데이터를 한 형식이나 구조에서 다른 형 식이나 구조로 변환
- 원본 데이터와 대상 데이터간에 필요한 데이터 변경 내용을 기반으로 데이터 변 환이 간단하거나 복잡 할 수 있음
- 데이터 변환은 일반적으로 수동 및 자동
- 단계가 혼합되어 수행
- 데이터 변환에 사용되는 도구 및 기술은 변환되는 데이터의 형식, 구조, 복잡성 및 볼륨에 따라 크게 다를 수 있음



1. 데이터 변환(Data Transformation)

주요 데이터 변환 기법

구분	기법	설명
정규화	L1, L2 Norm	벡터 크기를 1로 맞춤
스케일링	Min-Max, Standard	값의 범위를 맞추는 작업
로그 변환	log(x)	큰 값의 영향 완화, 분포 조정
루트 변환	sqrt(x)	약한 비선형 효과 조정
Box-Cox 변환	통계적 분포를 정규분포에 가깝게	
원-핫 인코딩	pd.get_dummies()	범주형 → 이진 벡터
라벨 인코딩	LabelEncoder	범주형 → 정수 값

2. 정규화

정규화(Normalization)?

- 정규화는 데이터의 값을 특정 범위로 조정하는 프로세스. 정규화는 속성값으로 -1.0 ~ 1.0과 같이 정해진 구간 내에 들도록 하는 기법

  - 일반적으로 데이터를 [0, 1] 또는 [-1, 1] 범위로 맞춤. 최소-최대 정규화(Min-Max Normalization)를 사용하여 최솟값을 0, 최댓값을 1로 변환하는 방법이 있다.
  - 최단 근접 분류와 군집화와 같은 거리 측정 등을 위해 특히 유용

2. 정규화

정규화 종류

- 최소-최대 정규화min-max normalization
  - 원본 데이터에 대하여 선형 변환 수행
  - 속성 A에 대한 최소값과 최대값을 $minA$와 $maxA$라고 가정
  - A의 값은 다음 계산식에 의해 v를 구간 에서의 값 v'으로 사상

$$v' = \frac{v - min_A}{max_A - min_A}\left(max'_A - min'_A\right) + min'_A$$

  - 최소-최대 정규화는 원본 데이터 값들 간의 관계를 보존
  - 정규화를 위한 입력이 A에 대한 원본 데이터 구간에서 벗어난 경우는 범위 초과out-of-bounds 오류 발생

2. 정규화

정규화 종류

- 구간화(Min-Max)
  - 최대값과 최소값을 사용하여 원 데이터의 최소값을 0, 최대값을 1로 만드는 방법이다. 여기에 100을 곱하여 지표관리 등 다양한 곳에 활용하기도 한다.

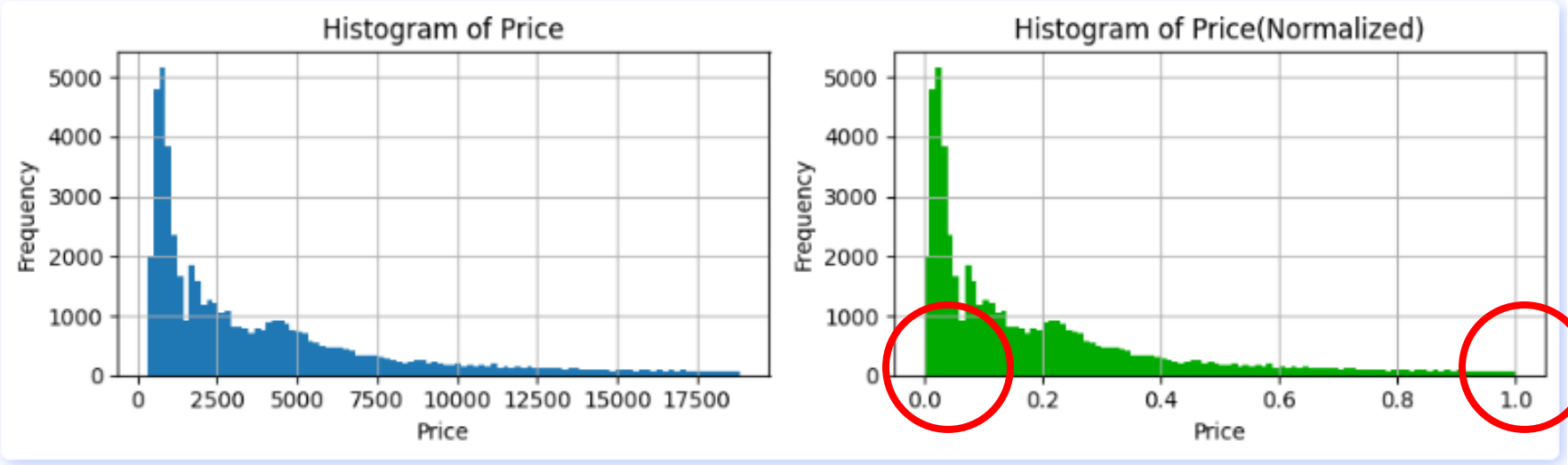
$$MinMax(x) = \frac{x - min(x)}{max(x) - min(x)}$$

  - 여기서 최소값을 빼는 것은 최소값을 0으로 만들며 범위(max - min)를 나누는 것은 범위를 1로 만들기 때문에 최종적으로 최대값이 1이 된다.

2. 정규화

정규화 종류

- 구간화(Min-Max)
  - 구간화 전과 후의 분포를 비교하면 다음과 같다.



2. 정규화

정규화 종류

- 소수 척도화decimal scaling
  - 속성 A 값들의 소수점을 이동해서 정규화
  - 이동되는 소수점의 수는 A의 최대 절대값에 의존
  - $Max\ v' < 1$을 만족하는 가장 작은 정수를 j라고 가정

  - A의 값 v는 다음 계산식에 의해 v'로 사상

$$v' = \frac{v}{10^j}$$

2. 정규화

정규화 예

- 아래 예제는 sklearn 라이브러리의 MinMaxScaler를 사용하여 최소-최대 정규화(Min-Max Normalization)를 수행하는 예시.
  - 최소-최대 정규화는 데이터를 특정 범위(일반적으로 0과 1 사이)로 변환하는 정규화 기법 중 하나. 주어진 데이터를 0과 1 사이의 값으로 변환.
  - 여기서 data는 2차원 리스트로, 각 행은 하나의 특성을 갖고 있다. MinMaxScaler를 사용하여 fit_transform() 메서드를 호출하여 데이터를 정규화.

2. 정규화

정규화 예

```
from sklearn.preprocessing import MinMaxScaler

# 데이터 생성 (예시)
data = [[10], [5], [3], [2], [8]]

# 최소-최대 정규화 적용
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data)
print("정규화된 데이터:")
print(scaled_data)
>>
정규화된 데이터:
[[1.   ]
 [0.375]
 [0.125]
 [0.   ]
 [0.75 ]]
```

$$MinMax(x) = \frac{x - min(x)}{max(x) - min(x)}$$

1. $[10] \rightarrow \frac{10-2}{10-2} = 1$
2. $[5] \rightarrow \frac{5-2}{10-2} = \frac{3}{8}$
3. $[3] \rightarrow \frac{3-2}{10-2} = \frac{1}{8}$
4. $[2] \rightarrow \frac{2-2}{10-2} = 0$
5. $[8] \rightarrow \frac{8-2}{10-2} = \frac{6}{8} = \frac{3}{4}$

3. 표준화

표준화(Standardization)

- 평균과 표준편차를 사용하여 평균이 0, 표준편차를 1로 만드는 방법. 표준화는 데이터를 평균이 0이고 표준편차가 1인 분포로 변환하는 것을 의미. 각 데이터에서 평균을 빼고 표준편차로 나누어 변환.
- 흔히 z-scoring 이라고 하기도 한다.

$$Standardization(x) = \frac{x - mean(x)}{std(x)}$$

- 여기서 평균을 빼는 것은 데이터의 중심을 0으로 옮기는 것이며 표준편차를 나누는 것은 자료의 편차를 1로 만드는 것이 된다.

3. 표준화

표준화(Standardization)

- Z-score

Z is the Z-score,
X is the value,
μ is the mean of the population,
σ is the standard deviation of the population.

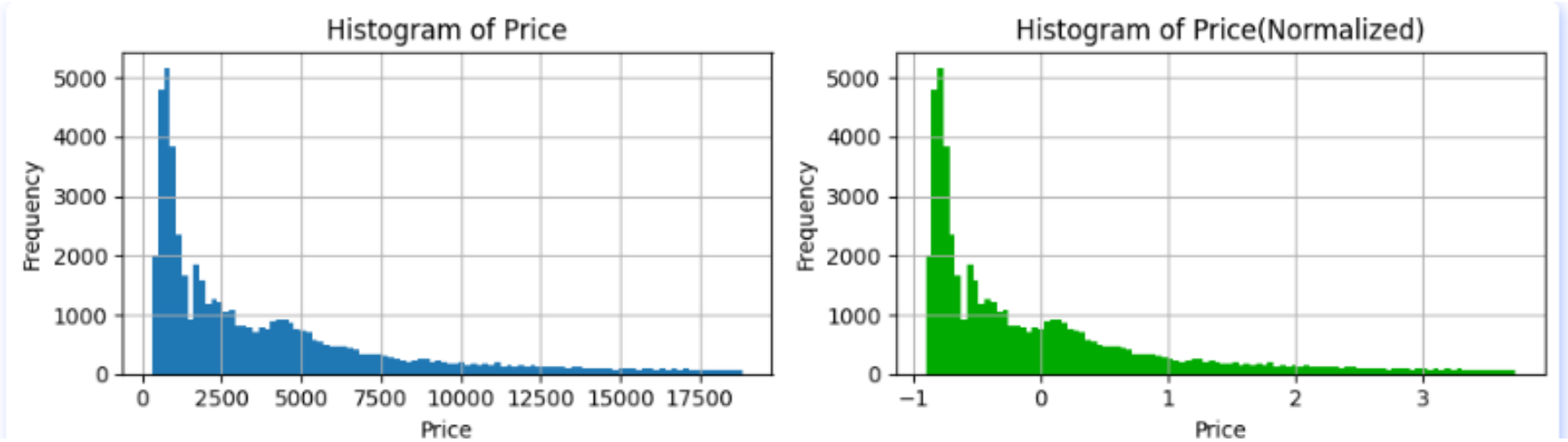
$$Z = \frac{X - \mu}{\sigma}$$

- 속성 A에 대한 값을 A의 평균과 표준편차를 기초로 정규화하는 방법
- Z-score 정규화는 속성 A의 실제 최소값과 최대값이 알려져 있지 않거나, 최소-최대 정규화에 큰 영향을 주는 이상치가 존재할 때 유용

3. 표준화

표준화(Standardization)

- 표준화(Standardization) 전과 후의 분포를 비교하면 다음과 같다.



3. 표준화

표준화(Standardization)

- 주어진 코드는 sklearn 라이브러리의 StandardScaler를 사용하여 데이터를 표준화(Standardization)하는 예시.
  - 표준화는 데이터의 평균을 0으로, 표준편차를 1로 만들어 데이터를 정규분포에 가깝게 만드는 작업. 따라서, 주어진 코드에서 data는 2차원 리스트로, 각 행은 하나의 특성을 나타냄.
  - StandardScaler를 사용하여 fit_transform() 메서드를 호출하여 데이터를 표준화.
  - 실행된 결과인 standardized_data는 표준화된 데이터를 나타내며, 각 값이 평균이 0이고 표준편차가 1인 정규분포에 가까워진 것을 확인할 수 있다.

3. 표준화

표준화(Standardization)

```
from sklearn.preprocessing import StandardScaler

# 데이터 생성 (예시)
data = [[10], [5], [3], [2], [8]]

# 표준화 적용
scaler = StandardScaler()
standardized_data = scaler.fit_transform(data)
print("표준화된 데이터:")
print(standardized_data)
>>
표준화된 데이터:
[[ 1.46341823]
 [-0.19955703]
 [-0.86474714]
 [-1.19734219]
 [ 0.79822813]]
```

$$Z = \frac{X - \mu}{\sigma}$$

평균 계산:

$$\text{평균} = \frac{10+5+3+2+8}{5} = \frac{28}{5} = 5.6$$

표준편차 계산:

$$\text{표준편차} = \sqrt{\frac{\sum (X_i - \text{평균})^2}{N}}$$

$$= \sqrt{\frac{(10-5.6)^2+(5-5.6)^2+(3-5.6)^2+(2-5.6)^2+(8-5.6)^2}{5}}$$

$$= \sqrt{\frac{20.8+0.36+10.96+14.44+4.84}{5}}$$

$$= \sqrt{\frac{51.4}{5}}$$

$$= \sqrt{10.28}$$

$$\approx 3.21$$

이제 Z 점수를 계산하여 데이터를 표준화합니다:

1. $[10] \rightarrow \frac{10-5.6}{3.21} \approx 1.37$
2. $[5] \rightarrow \frac{5-5.6}{3.21} \approx -0.19$
3. $[3] \rightarrow \frac{3-5.6}{3.21} \approx -0.81$
4. $[2] \rightarrow \frac{2-5.6}{3.21} \approx -1.12$
5. $[8] \rightarrow \frac{8-5.6}{3.21} \approx 0.75$

4. 수치형 데이터 이산화

수치형 데이터 이산화(Numeric Data Discretization)

- 이산화(Discretization) 란, 연속적인 수치형 데이터를 구간으로 나누어 이산적(불연속적)인 값으로 변환하는 것을 의미.
  - 즉, 실수 → 범주형 변수로 바꾸는 과정.
  - 예시: 나이(age)를 이산화하기

원본 값 (연속)	이산화 후 (범주)
23	청년
42	중년
65	노년

4. 수치형 데이터 이산화

주요 이산화 방법

방법	설명	예시
균등 간격 분할 (Equal-width)	전체 범위를 동일한 폭으로 나눔	[010), [1020), ...
균등 빈 분할 (Equal-frequency)	각 구간에 데이터 수가 같도록 나눔	사분위수 기준
클러스터 기반 (KMeans, DecisionTree)	데이터 특성 기반 자동 구간화	정보이득 기반
도메인 기반 수동 정의	도메인 지식을 반영하여 구간 정의	나이: 청년/중년/노년

4. 수치형 데이터 이산화

수치형 데이터 이산화 예

```
import pandas as pd

# 예시 데이터
ages = pd.Series([18, 25, 33, 45, 60, 70])

# 구간 정의
bins = [17.948, 35.333, 52.667, 70.0]
labels = ['청년', '중년', '노년']

# 이산화 수행
age_groups = pd.cut(ages, bins=bins, labels=labels)

# 결과 출력
df = pd.DataFrame({'나이': ages, '연령대': age_groups})
print(df)
```

bins: 경계값 리스트 (왼쪽 열림, 오른쪽 닫힘: (a, b])

```
   나이 연령대
0  18  청년
1  25  청년
2  33  청년
3  45  중년
4  60  노년
5  70  노년
```

『1-6』
데이터 준비(Data Preparation)

Data Integration (통합)

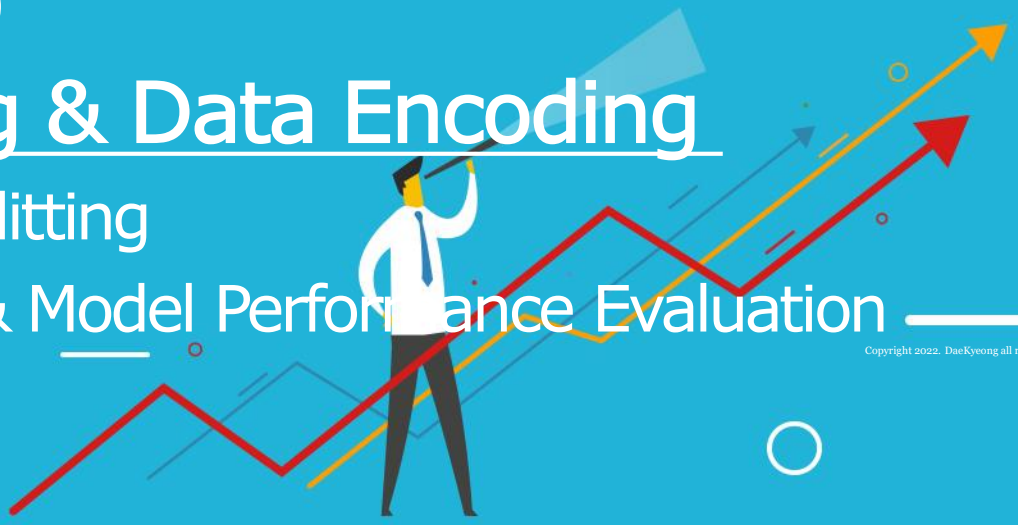
Data Reduction (축소)

Data Transformation (변환)

Feature Engineering & Data Encoding

Cross Validation & Data Splitting

Data Quality Assessment & Model Performance Evaluation



Copyright 2022. DaeKyeong all rights reserved

개요

학습목표

- 이 워크샵에서는 특성 공학(Feature Engineering)과 데이터 인코딩(Data Encoding) 에 대해 알 수 있습니다.

눈높이 체크

- 특성 공학(Feature Engineering)과 데이터 인코딩(Data Encoding) 에 대해 들어보셨나요?

1. Feature Engineering

특성 공학(Feature Engineering)?

- 특성 공학(Feature Engineering)은 머신러닝 모델에 입력으로 제공될 특성(또는 변수)을 만들거나 변형하는 과정.

  - 이를 통해 모델의 성능을 향상시키고, 데이터로부터 더 유용한 정보를 추출할 수 있다.
  - 특성 공학은 모델의 성능을 향상시키고 데이터의 정보를 최대한 활용하기 위해 매우 중요한 단계. 하지만 도메인 지식과 실험을 통한 탐색이 필요. 데이터의 특성을 잘 이해하고, 문제에 적합한 특성 공학 기법을 적용하는 것이 중요.

1. Feature Engineering

특성 공학(Feature Engineering)?

- 일반적인 특성 공학의 목표는 다음과 같다.
  - 새로운 특성 생성: 기존의 특성을 기반으로 새로운 특성을 만들어내는 것입니다. 예를 들어, 날짜 데이터에서 요일, 월, 계절 등을 추출하거나, 길이에 관련된 특성을 만들거나, 텍스트 데이터에서 단어 수 또는 패턴을 추출하는 등이 있습니다.
  - 특성 변형: 기존의 특성을 변형하여 새로운 관점에서 데이터를 표현합니다. 예를 들어, 로그, 제곱근, 표준화, 정규화 등을 사용하여 데이터의 분포를 조정할 수 있습니다.
  - 차원 축소: 고차원의 데이터를 저차원으로 변환하여 더 간결하고 효과적인 데이터를 생성합니다. 주성분 분석(PCA), t-SNE 등의 기법을 사용하여 차원을 축소할 수 있습니다.
  - 외부 데이터 사용: 외부 데이터를 활용하여 새로운 특성을 추가하는 것도 중요한 특성 공학의 한 부분입니다. 예를 들어, 지리적 데이터를 활용하여 거리 기반 특성을 만드는 등이 있습니다.

1. Feature Engineering

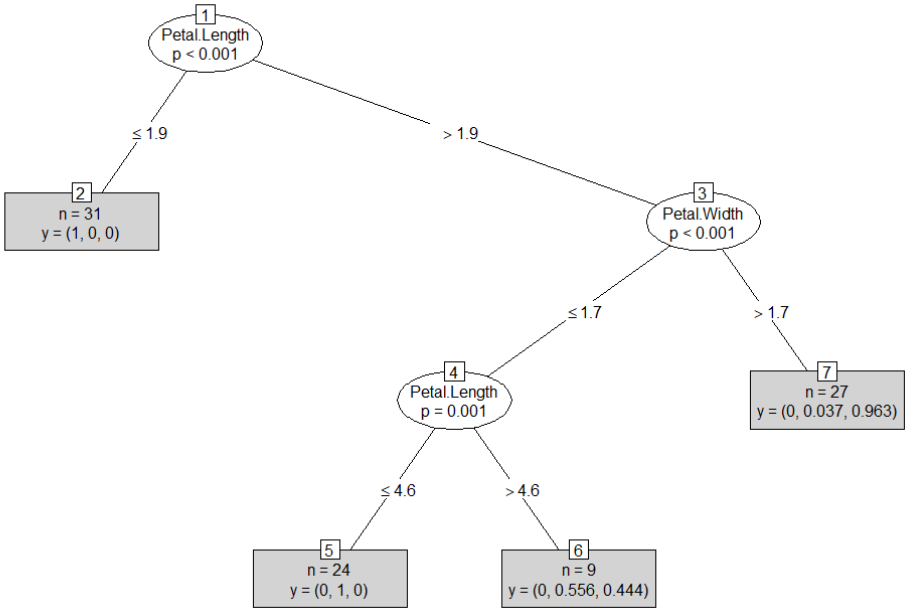
특징 값의 종류

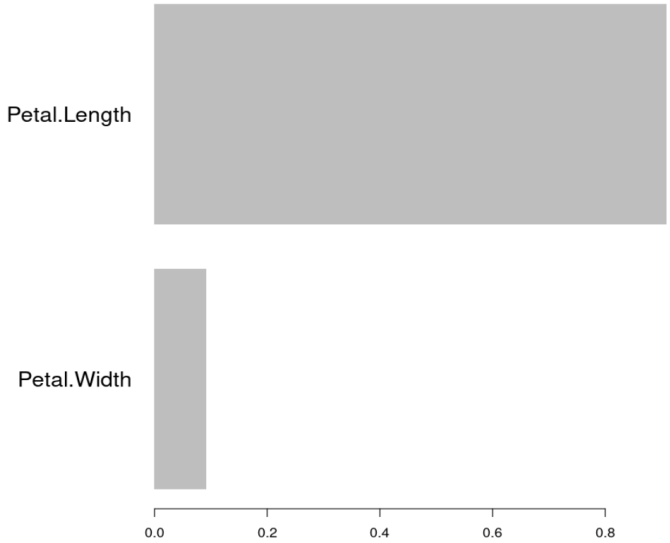
- 수치형 특징
  - 예) iris의 네 개 특징은 실수
  - 거리 개념이 있음
  - 실수 또는 정수 또는 이진값
- 범주형 특징
  - 학점, 수능 등급, 혈액형, 지역 등
  - 순서형: 학점, 수능 등급 등
    - 거리 개념이 있음. 순서대로 정수를 부여하면 수치형으로 취급 가능
  - 이름형
    - 혈액형, 지역 등으로 거리 개념이 없음
    - 보통 원핫one-hot 코드로 표현. 예) A형(1,0,0,0), B형(0,1,0,0), O형(0,0,1,0), AB형(0,0,0,1)

1. Feature Engineering

특징 공간에서 데이터 분포

- petal width(수직 축)에 대해 Setosa는 아래쪽, Virginica는 위쪽에 분포→ petal Length 특징은 분별력discriminating power이 뛰어남
- sepal width 축은 세 부류가 많이 겹쳐서 분별력이 낮음
- 전체적으로 보면, 세 부류가 3차원 공간에서 서로 다른 영역을 차지하는데 몇 개 샘플은 겹쳐 나타남





2. Data Encoding

원핫 인코딩(One-Hot Encoding):

- 원핫 인코딩(One-Hot Encoding)은 범주형 데이터를 머신러닝 알고리즘이 이해할 수 있는 형태로 변환하는 방법 중 하나. 주로 범주형 변수의 각 범주를 새로운 이진 특성으로 변환.
- 파이썬에서 pandas나 scikit-learn을 사용하여 원핫 인코딩을 수행. 아래는 pandas를 사용하여 원핫 인코딩을 수행하는 예제 코드입니다.

2. Data Encoding

원핫 인코딩(One-Hot Encoding):

```
import pandas as pd

# 샘플 데이터셋
data = {
    'color': ['red', 'blue', 'green', 'red', 'yellow']
}

# DataFrame 생성
df = pd.DataFrame(data)

# 원핫 인코딩 적용
df_encoded = pd.get_dummies(df['color'], prefix='color')

# 변환된 데이터 확인
print("변환된 데이터:")
print(df_encoded)
```

>>

변환된 데이터:

	color_blue	color_green	color_red	color_yellow
0	False	False	True	False
1	True	False	False	False
2	False	True	False	False
3	False	False	True	False
4	False	False	False	True

2. Data Encoding

원핫 인코딩(One-Hot Encoding):

- 위 코드에서 pd.get_dummies() 함수를 사용하여 'color' 열을 원핫 인코딩. 각 범주는 새로운 이진 특성으로 변환되었으며, 변환된 데이터셋은 df_encoded에 저장되어 출력.
- 실제로는 범주형 변수의 종류와 데이터에 따라 적절한 원핫 인코딩 방식을 선택. 인코딩된 데이터는 기존 변수보다 차원이 증가할 수 있으므로, 데이터 크기와 모델 성능에 영향을 미칠 수 있음.

2. Data Encoding

데이터 인코딩(Data Encoding)

- 데이터 인코딩은 주로 범주형 데이터를 모델링하거나 분석하기 쉬운 형태로 변환하는데 사용.

  - 라벨 인코딩 (Label Encoding): 범주형 변수를 숫자형으로 변환하는 기법입니다. 각 범주형 변수의 고유한 값들을 숫자로 매핑하여 변환합니다. 이를 통해 모델이 이해할 수 있는 형태로 변환하지만, 값 사이의 순서나 관계를 시사하는 것은 아닙니다. 예를 들어, {고양이, 개, 새}와 같은 범주형 변수를 {0, 1, 2}와 같은 숫자로 변환합니다.
  - 더미 변수화 (Dummy Variable Creation 또는 One-Hot Encoding):범주형 변수를 이진형 변수로 변환하는 기법입니다. 각 범주에 대한 새로운 이진형 변수(0 또는 1)를 생성합니다. 원-핫 인코딩이라고도 불리며, 각 범주를 별도의 열로 만들어 해당 범주에 해당하는 열은 1로 표시하고, 나머지 열은 0으로 표시합니다. 이 방법은 범주 간의 관계나 순서를 고려하지 않고, 각 범주를 독립적으로 처리할 수 있도록 합니다.

2. Data Encoding

데이터 인코딩(Data Encoding)

- 라벨 인코딩은 순서가 있는 범주형 데이터에 적합하며, 더미 변수화는 순서가 없는 범주형 데이터에 적합합니다. 데이터와 모델의 요구 사항에 따라 적절한 데이터 인코딩 방법을 선택하여 사용해야 합니다.

2. Data Encoding

라벨 인코딩

- 라벨 인코딩은 범주형 데이터를 숫자로 변환하는 과정.
  - 파이썬에서 LabelEncoder를 사용하여 라벨 인코딩을 수행할 수 있다. sklearn.preprocessing 모듈에 포함되어 있다.
  - 아래 코드는 categories 리스트의 각 범주를 라벨 인코딩하여 숫자로 변환. fit_transform 메서드를 사용하여 변환을 수행하고, 변환된 결과를 encoded_labels에 저장. 결과는 각 범주에 대해 할당된 숫자.

```
from sklearn.preprocessing import LabelEncoder

# 범주형 데이터 예시
categories = ['고양이', '개', '고양이', '새', '개']

# LabelEncoder 객체 생성
label_encoder = LabelEncoder()

# 라벨 인코딩 수행
encoded_labels = label_encoder.fit_transform(categories)
print(encoded_labels)

>>
[1 0 1 2 0]
```

2. Data Encoding

더미 변수화 (Dummy Variable Creation 또는 One-Hot Encoding)

- 파이썬에서 더미 변수화 또는 원-핫 인코딩을 수행하는 가장 흔한 방법은 pandas 라이브러리의 get_dummies() 함수를 사용하는 것. 이 함수를 사용하여 범주형 변수를 이진형 변수로 변환할 수 있다.
  - 아래 코드는 pet 열을 가진 데이터프레임을 생성하고, pd.get_dummies() 함수를 사용하여 pet 열을 더미 변수로 변환. concat() 함수를 사용하여 원본 데이터프레임과 더미 변수를 합쳐서 df_encoded라는 새로운 데이터프레임을 생성.
  - 더미 변수화를 통해 각 범주가 이진형 변수로 변환되며, 해당 범주에 속할 경우 1로 표시되고 속하지 않을 경우 0으로 표시. 이렇게 변환된 데이터프레임은 머신러닝 모델에 바로 사용될 수 있다.

2. Data Encoding

더미 변수화

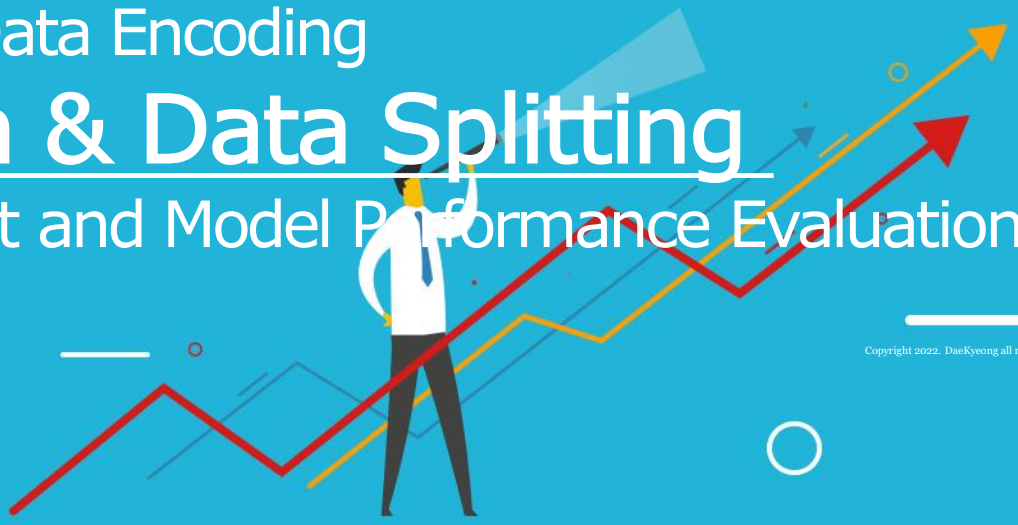
```
import pandas as pd

# 범주형 데이터 예시
data = {'pet': ['고양이', '개', '새', '고양이', '개']}
df = pd.DataFrame(data)

# 더미 변수화 (원-핫 인코딩)
dummy_variables = pd.get_dummies(df['pet'])

# 원본 데이터프레임과 더미 변수를 합치기
df_encoded = pd.concat([df, dummy_variables], axis=1)
print(df_encoded)
>>
```

```
    pet     개    고양이     새
0  고양이  False   True  False
1    개   True  False  False
2    새  False  False   True
3  고양이  False   True  False
4    개   True  False  False
```

```
      pet  고양이  개  새
0  고양이    1  0  0
1    개    0  1  0
2    새    0  0  1
3  고양이    1  0  0
4    개    0  1  0
```

『1-6』
데이터 준비(Data Preparation)

- Data Integration (통합)
- Data Reduction (축소)
- Data Transformation (변환)
- Feature Engineering & Data Encoding
- **Cross Validation & Data Splitting**
- Data Quality Assessment and Model Performance Evaluation



Copyright 2022. DaeKyeong all rights reserved

Syllabus

학습목표

- 이 워크샵에서는 Cross Validation & Data Splitting 에 대해 알 수 있습니다.

눈높이 체크

- Cross Validation & Data Splitting 을 알고 계신가요?

1. 교차 검증 (Cross Validation, CV)

교차검증?

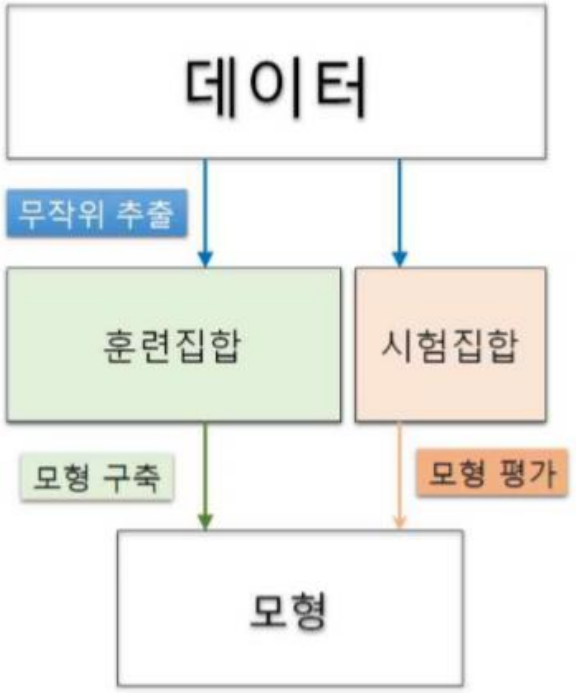
- 데이터 전처리 파이프라인을 만들고 모델을 훈련한 다음 교차검증으로 평가.
  - 처음 모델을 훈련하고 어떤 성능 지표(정확도, 제곱 오차 등)를 사용하여 동작하는 계산, 그러나 이러한 방법은 모델을 훈련한 데이터로 모델을 평가함으로 원하는 것과 다름.
  - 따라서 훈련 데이터가 아닌 이전에 본 적 없는 새로운 데이터에서 잘 동작하는 지를 평가해야 함
  - 데이터를 여러 개의 **fold(조각)**으로 나눈 후, 여러 번 훈련·평가를 반복하여 평균 성능을 평가하는 방식.

  - K-Fold Cross Validation
    - 데이터를 K등분하고, 그 중 1개는 테스트용, 나머지는 학습용으로 사용
    - K번 반복하면서 테스트 데이터를 바꿔가며 평균 성능 측정
    - 이외 에도 StratifiedKFold, Leave-One-Out, ShuffleSplit 등 다양한 교차 검증이 있음

1. 교차 검증 (Cross Validation, CV)

홀드아웃 교차 방법 개념

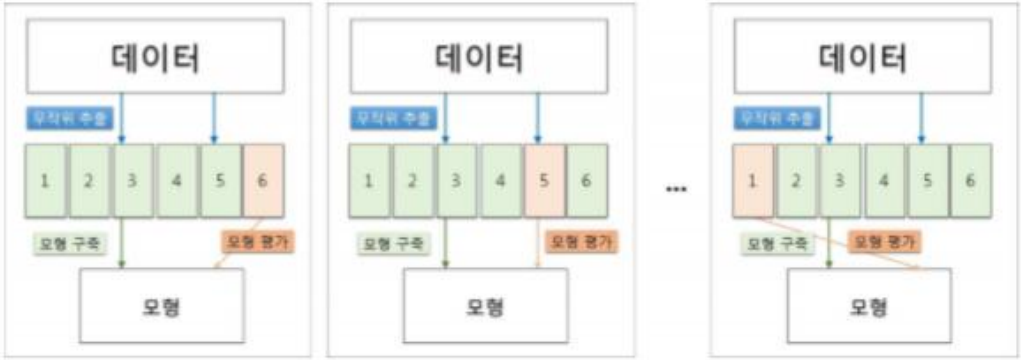
- 개념
  - 데이터 집합을 서로 겹치지 않는 훈련 집합(training set)과 시험 집합(test set)으로 무작위로 구분한 후, 훈련 집합을 이용하여 분석 모형을 구축하고 시험 집합을 이용하여 분석 모형의 성능을 평가하는 기법이다(P. Tan, M. Steinbach, and V. Kumar, 2007). 훈련 집합과 시험 집합의 비율은 50-50, 70-30 등 사용자에 의해서 결정된다.



1. 교차 검증 (Cross Validation, CV)

다중 교차 방법

- 개념
  - 데이터 집합을 무작위로 동일 크기를 갖는 k개의 부분 집합으로 나누고, 그중 1개를 시험 집합으로, 나머지 k-1개를 훈련 집합으로 선정하여 분석 모형을 평가한다(P. Tan, M. Steinbach, and V. Kumar, 2007). 이러한 방식으로 모든 부분 집합들을 시험 집합으로 정확히 1회씩 선정하여 총 k번 반복한다.
- 특징
  - 이 방법은 홀드아웃 교차 검증과는 달리 모든 데이터 집합을 훈련 집합 및 시험 집합으로 사용하기 때문에 분석 모형의 평가 결과가 편향되지 않는다는 장점이 있다. [그림]은 k=6인 다중 교차 검증의 개념을 나타낸다.



1. 교차 검증 (Cross Validation, CV)

기타

- Random subsampling:Holdout 방식 반복
  - K개의 부분 데이터 셋 사용: 각 데이터 셋은 랜덤
  - 최종 성능은 각 실험 성능의 평균으로 도출
- stratified sampling
  - 각 클래스로부터 일정 비율 샘플 추출
  - 전체 데이터에서 무작위로 추출할 경우 표본이 특정 클래스에 편중될 수 있기 때문에 사용
- Botstrap
  - 중복 추출 허용

1. 교차 검증 (Cross Validation, CV)

교차검증?

- 예시 (5-Fold Cross Validation)

Fold	Train 데이터	Test 데이터
1	Fold 2~5	Fold 1
2	Fold 1,3~5	Fold 2
…	…	…
5	Fold 1~4	Fold 5

2. Data Splitting

데이터 분할(Data Splitting)?

- 데이터 분할(Data Splitting)은 기계 학습에서 중요한 단계 중 하나. 데이터를 학습용과 테스트용으로 나누는 것으로, 모델을 훈련하고 테스트하기 위해 데이터셋을 두 부분으로 분할하는 작업.

- 학습용 데이터와 테스트용 데이터 분리:
  - 일반적으로 데이터의 대부분(70-80%)을 학습에 사용하고, 나머지 부분(20-30%)을 모델의 성능 평가를 위한 테스트에 사용.
  - 최근에는 전체 데이터를 **훈련용(Train)**과 **검증/테스트용(Test/Validation)**으로 나눔.

2. Data Splitting

데이터 세트 준비 및 분할

- 일반적으로, 머신 러닝 기반 데이터 분석 진행 시, 특히 지도학습 기반 모델 적용을 할 때는 전체 데이터 세트를 사용하여 한꺼번에 분석하지 않고, 학습용 데이터 세트와 평가용(테스트) 데이터 세트로 분할하여 분석을 진행.
- 분석하고자 하는 목적 및 데이터 세트 특성에 따라 머신 러닝 기법 적용을 위한 훈련 데이터 세트와 테스트 데이터 세트 분할 기준을 판단할 수 있다. 평가 데이터 세트를 사용하는 목적은 '미지의 데이터를 예측하는 능력, 즉 머신 러닝 기반 분석모델의 일반화 능력을 측정하고 성능을 향상하기 위함이다.'라고 할 수 있다.
- 그러나 일반적으로 훈련 데이터와 유사하면서 목적에 적합한 평가 데이터를 별도로 구하기는 쉽지 않으므로, 모델링을 위해 주어진 데이터를 학습용 데이터와 평가용 데이터로 분할해서 머신 러닝 기법을 적용하게 되는 것.
- 해결하고자 하는 이슈와 적용할 기법에 따라 교차검증 필요성을 판단하여, 훈련 데이터 세트와 검증 데이터 세트를 분할하고, 적합한 교차검증 K값을 결정할 수 있다

2. Data Splitting

데이터 세트 준비 및 분할

- 머신 러닝 기반 데이터 분석을 진행함에 있어서, 머신 러닝 기법이 주로 하는 역할은 주어 진 데이터 세트를 학습하여 최적의 모수(파라미터)를 도출하는 것과 이를 바탕으로 특정 설명변수(혹은 특성(Feature))가 주어졌을 때 목적변수 (혹은 반응변수)의 값을 예측하는 과정이라고 할 수 있습니다.
- 그런데 주어진 훈련 데이터 세트에 포함된 데이터는 엄밀히 말하면 '우연에 의해 얻어진 값'이라고 볼 수 있으므로 새로운 목적변수(혹은 반응변수) 등의 값을 예측하기 위해 얻어지는 신규 데이터 세트는 원래의 훈련 데이터 세트와 동일한 데이터가 아닙니다. 그러므로, 훈련데이터에서 나타난 패턴들과 신규 데이터의 패턴이 정확하게 일치할 가능성은 상당히 낮습니다.

2. Data Splitting

데이터 세트 준비 및 분할

- 따라서 머신러닝 기반 모델을 학습하는 데 있어서, 훈련 데이터 세트가 가지고 있는 특성을 너무 많이 반영하게 되면 훈련 데이터 세트의 패턴만 잘 표현하게 되는 '과적합(Overfitting)'이 발생하게 되고, 새로운 데이터가 주어졌을 때 정확하게 예측할 수 있는 '일반화(Generalization)' 능력은 오히려 떨어지게 됩니다.
- 그래서 이러한 현상을 방지하고자 일반적으로 데이터 세트를 훈련용 데이터 세트와 평가용 데이터 세트로 분할하고 훈련용 데이터 세트로 학습한 머신 러닝 모델이 평가용 데이터 세트의 목적변수(혹은 반응변수)를 얼마나 정확하게 예측하는지를 측정하여 이러한 기준치를 모델 성능의 평가 기준으로 삼게 되는 것입니다.

2. Data Splitting

데이터 세트 분할 방법 및 절차

1. 일정 비율로 학습용과 평가용 세트로 데이터 분할

- 데이터의 일부를 훈련 데이터, 나머지를 평가데이터로 분리합니다. 특별한 경우가 아니라면 일반적으로 학습용과 평가용 데이터 각각의 분할은 전체 데이터에서 랜덤하게 특정 비율로 학습용 데이터를 추출하고, 학습용 데이터에 사용되지 않은 나머지 데이터를 평가용 데이터로 취하는 방법을 따릅니다. 이때 훈련 데이터와 평가데이터를 분할하는 비율은 정해진 원칙이 있는 것은 아니나, 모델을 훈련시키는 과정 자체에 더 많은 비중을 할당합니다.
- 일반적으로 훈련 데이터를 60%~80%, 평가데이터를 40%~20% 정도로 할당합니다. 그러나 절대적인 기준은 아니며, 실무 상황에서는 분석의 목적이나 연구수행자의 판단과 경험을 통해 분할 비율을 정하게 됩니다. 다만 데이터 세트 분할 시 중요한 점은 실제 훈련된 모델의 성능은 학습용 데이터 세트 크기가 작아질수록 나빠지게 되므로, 너무 많은 데이터를 평가용 데이터로 분할하는 것은 최종 성능에 오히려 나쁜 영향을 끼칠 수 있다는 점입니다.

2. Data Splitting

데이터 세트 분할 방법 및 절차

2. 학습(훈련) 데이터로부터 머신러닝 모델링 수행

- 머신 러닝 모델링을 수행할 때 여러 가지 기법을 적용하여 기법 간의 성능을 비교할 수도 있고, 동일 기법 내에서도 추정방법을 변경하거나 파라미터를 다양하게 변경하는 등의 과정을 거치게 됩니다. 여기서 모델링 성능에 대한 보다 정교한 검증을 위해 교차검증 (Cross-Validation) 방법을 수행하는 경우도 있습니다.
- 교차검증은 훈련 데이터를 통한 모델링 훈련 시 훈련 데이터 내에서 별도의 검증 데이터(Validation Data)를 할당하여 모델링 및 평가를 반복하는 것으로서, 앞서 1단계에서 분할한 평가 데이터가 학습모델링에 사용되지 않는 반면, 교차검증 데이터는 학습모델링에 사용된다는 점이 다릅니다.

2. Data Splitting

데이터 세트 분할 방법 및 절차

3. 평가 데이터를 이용한 모델 성능 평가

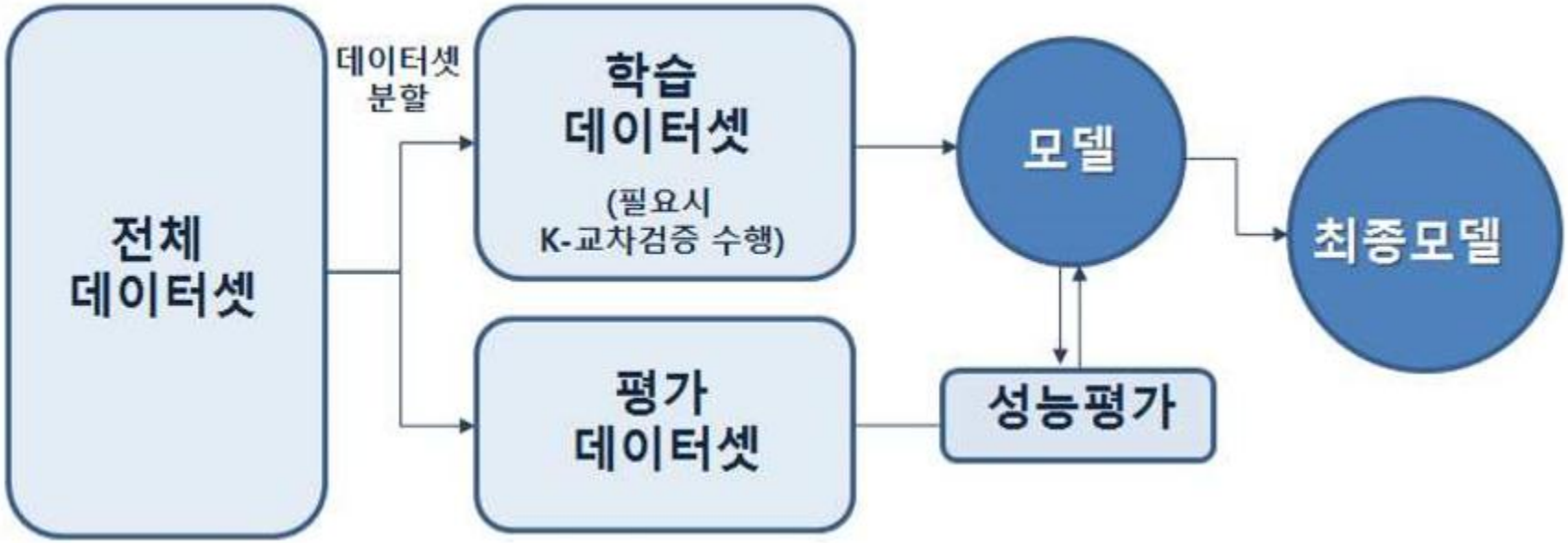
- 2단계에서 만들어진 모델에 평가 데이터를 적용해 성능을 평가합니다. 만일 성능이 만족스럽지 않다면, 앞의 2단계로 다시 돌아가게 됩니다.
- 여기서 평가 데이터는 원래의 전체 데이 터 세트로부터 최초 분리해낸 뒤, 모델링 과정에서 이용되지 않다가 2단계 등을 통해 모델이 만들어지고 난 뒤, 해당 모델의 성능을 평가하기 위해 3단계에서 사용됩니다.
- 이런 점에서 모델링 과정 자체에서 검증의 목적으로 반복적으로 사용되는 검증 데이터와는 그 활용목적이 다르다고 할 수 있습니다.

2. Data Splitting

데이터 세트 분할 방법 및 절차

4. 최종 모델 결과 제출

- 3단계에서 평가 데이터를 이용하여 수행한 성능 평가 및 예측결과가 기준치에 부합하거나, 목적에 적합하다고 판단될 경우 분석 모델링 과정을 종료하고, 최종 분석결과를 제출하게 됩니다.
- 훈련 데이터와 평가 데이터 분할 통한 머신러닝 모델링 절차



2. Data Splitting

고객 이탈 예측 모델 A~Z 시나리오

1. 전체 데이터셋 확보
- 고객 10,000명의 데이터 수집
- 주요 변수:
  - 고객 특성: 나이, 지역, 가입기간, 멤버십 여부
  - 이용 행동: 최근 접속 횟수, 월평균 결제액, 불만 접수 건수
  - 타겟 변수: 이탈 여부 (1=이탈, 0=유지)

2. 데이터셋 분할
- 학습 데이터셋 (70%): 7,000명 → 모델 학습용
- 평가 데이터셋 (30%): 3,000명 → 성능 검증용

필요시 K-겹 교차검증(K-fold CV) 수행 (예: K=5)
→ 데이터가 편향되지 않았는지, 과적합이 없는지 확인

2. Data Splitting

고객 이탈 예측 모델 A~Z 시나리오

3. 모델 학습

- 모델 후보군:
  - 로지스틱 회귀
  - 랜덤포레스트
  - XGBoost
- 학습 데이터셋으로 훈련 후 교차검증 성능 비교

예:

- 로지스틱 회귀: 정확도 85%
- 랜덤포레스트: 정확도 91%
- XGBoost: 정확도 93%

2. Data Splitting

고객 이탈 예측 모델 A~Z 시나리오

4. 성능 평가

- 평가 데이터셋으로 최종 성능 검증
- XGBoost 결과:
  - 정확도(Accuracy): 92%
  - 정밀도(Precision): 0.89
  - 재현율(Recall): 0.90
  - F1-score: 0.895

사내 기준(정확도 ≥ 90%, 재현율 ≥ 0.85) 충족

5. 최종 모델 확정 및 제출

- XGBoost 모델을 최종모델로 선정
- 모델 파일(final_churn_model.pkl) 저장
- 보고서 작성:
  - 주요 변수 중요도: 최근 접속 횟수, 불만 접수 건수, 가입기간
  - 성능 요약: “D30 고객 이탈 예측 정확도 92%”
- 최종 분석 결과 제출

『1-6』
데이터 준비(Data Preparation)

- Data Integration (통합)
- Data Reduction (축소)
- Data Transformation (변환)
- Feature Engineering & Data Encoding
- Cross Validation & Data Splitting
- Data Quality Assessment and Model Performance Evaluation

Copyright 2022. DaeKyeong all rights reserved

Syllabus

학습목표

- 이 워크샵에서는 Data Quality Assessment and Model Performance Evaluation 에 대해 알 수 있습니다.

눈높이 체크

- Data Quality Assessment and Model Performance Evaluation 을 알고 계신가요?

1.Data Quality Validation

데이터 품질 검증(Data Quality Validation)이란?

- 데이터 품질 검증이란, 수집된 데이터가 정확하고, 일관되며, 완전하고, 유효한지 점검하는 과정.즉, “이 데이터를 믿고 써도 되는가?” 를 확인하는 작업

왜 데이터 품질 검증이 중요한가?

이유	설명
품질이 낮은 데이터	→ 잘못된 분석 결과 유발
머신러닝 성능 저하	→ 모델이 잘못된 패턴 학습
보고서 오류	→ 잘못된 비즈니스 의사결정

1.Data Quality Validation

주요 데이터 품질 검증 항목

- 데이터 분석의 목적을 달성하고, 최종 사용자의 기대를 만족하게 하기 위해 데이터가 확보하고 있어야 할 성질을 말한다. 데이터 품질을 보장하기 위해 만족하여야 할 요소로는 정확성(accuracy), 완전성(completeness), 적시성(timeliness), 일관성(consistency)이 존재한다(최상균, 전순천, 2013).

항목	설명	예시
정확성 (Accuracy)	실제 값과의 일치 여부	"성별"에 123 입력
완전성 (Completeness)	누락 없이 값이 채워졌는지	주소가 빈칸이면
일관성 (Consistency)	동일한 데이터가 동일하게 표현되는지	"남" / "남자" 혼재
유효성 (Validity)	정의된 포맷, 범위 안에 있는지	날짜: 2023-13-40
중복성 (Uniqueness)	중복된 레코드가 있는지	동일 고객이 두 번 입력
적시성 (Timeliness)	최신 데이터인지	2010년 고객 정보

1.Data Quality Validation

데이터 품질 검증을 위한 만족 요건

- 데이터 품질을 검증하기 위해서는 데이터 품질 요소들이 다음과 같은 요건을 만족해야 한다

1. 정확성
– 저장된 데이터는 대상을 올바로 나타내는 값을 가져야 함
2. 완전성
– 저장된 데이터는 대상을 올바로 나타내는 값을 가져야 함
3. 적시성
– 저장된 데이터는 대상을 올바로 나타내는 값을 가져야 함
4. 일관성
– 저장된 데이터는 대상을 올바로 나타내는 값을 가져야 함

1.Data Quality Validation

데이터 무결성

- 데이터 무결성(data integrity)이란 다수의 사용자가 데이터베이스에 접근하여 적재, 삽입, 삭제, 수정 등의 작업을 수행할 때 데이터가 불일치하지 않는 특성을 말한다.(___(2015.12.8.). 데이터 무결성 (Data Integrity). http://bongbonge.tistory.com/100에서 2016.7. 8. 검색). 데이터 무결성은 데이터 품질을 만족하는 데이터가 분석 목적을 달성하기 위해 공유되어 사용될 때 지켜져야 할 성질이다.

1.Data Quality Validation

데이터 무결성

- 데이터 무결성을 확보하기 위해서는 다음과 같은 필요 요건을 만족해야 한다.

1. 개체 무결성
– 기본 키(primary key)는 반드시 값을 가지며 그 값은 유일해야 함
2. 참조 무결성
– 외래 키(foreign key)값은 참조하는 테이블의 기본 키값 혹은 빈값 중 하나를 가져야 함
3. 속성 무결성
– 속성값은 지정된 데이터 형식을 만족하는 값을 가져야 함
4. 키 무결성
– 하나의 테이블에 적어도 하나의 키가 존재해야 함
5. 도메인 무결성
– 속성값은 미리 정의된 도메인 범위 안의 값을 가져야 함
6. 사용자 정의 무결성
– 모든 데이터는 업무 규칙(business rule)을 준수해야 함

1.Data Quality Validation

실제 수행 방법 예시 (Pandas 기반)

```
import pandas as pd

# 예시 데이터프레임
df = pd.DataFrame({
    '고객명': ['홍길동', '김철수', None],
    '생년': [1985, 2020, 1890],
    '이메일': ['hong@gmail.com', 'not-an-email', 'kim@naver.com']
})

# 평가 예시
print("결측치 확인:\n", df.isnull().sum())
print("생년 유효성 확인:\n", df[(df['생년'] < 1900) | (df['생년'] > 2024)])
print("이메일 유효성 확인:\n", df[~df['이메일'].str.contains('@')])
```

```
결측치 확인:
 고객명    1
생년      0
이메일     0
dtype: int64
생년 유효성 확인:
     고객명    생년            이메일
2  None  1890  kim@naver.com
이메일 유효성 확인:
    고객명    생년            이메일
1  김철수  2020  not-an-email
```

1.Data Quality Validation

개인정보보호

- **데이터 비식별화(de-identification)의 목적과 개념**

◦ **목적**

– 데이터 비식별화의 목적은 데이터에 포함된 개인정보를 삭제하거나 다른 정보로 대체하여 데이터 내에서 특정 개인을 식별하지 못하게 하기 위함이다(양현철, 신신애, 김진철, 2014).

1.Data Quality Validation

개인정보보호

◦ 개념

– 개인정보란 이름, 주민등록번호에서 DNA에 이르기까지 그것을 이용해 특정 개인을 식별할 가능성을 내포한 데이터를 말한다. 데이터 비식별화는 개인을 식별할 수 있는 잠재성을 가진 데이터를 식별할 수 없거나 식별하기 어려운 데이터로 가공하는 일련의 과정을 일컫는다. 비식별화는 SNS와 같은 개인정보를 포함하고 있는 데이터에 대한 분석이 증가하면서 그 중요성이 대두하고 있다. 미국 연방거래위원회(2012)는 보고서에서 다음과 같은 세 가지 비식별화 조치사항을 명시하였다.

- 소비자, 컴퓨터 또는 다른 장치와 결합할 수 있는 개인정보는 반드시 비식별화되어야 함
- 공개된 정보에 대해서는 재식별화를 시도하지 않아야 함
- 타 기업 등에 비식별화된 데이터 제공 시 데이터를 재식별화하지 않도록 계약상 명문화하도록 함

1.Data Quality Validation

개인정보보호

- 대표적인 비식별화 기법으로는 다음과 같은 것들이 있다
  - 가명처리(pseudonymisation)
    - 식별 가능한 변수값을 다른 값으로 대체
    - 예) 김치국, 38세, 수원 거주 -> 홍길동, 38세, 수원 거주
  - 총계처리(aggregation)
    - 개인정보 보호를 위해 데이터를 총합하거나 평균을 사용
    - 예) A 직원 연봉 4,500만, B 직원 연봉 5,200만, C 직원 연봉 4,600만
    - -> 평균 연봉 4,766만
  - 데이터 값 제거(data reduction)
    - 개인 식별에 유의한 변수값 제거
    - 예) 김치국, 38세, 수원 거주 -> 38세 남, 수원 거주

1.Data Quality Validation

개인정보보호

- 대표적인 비식별화 기법으로는 다음과 같은 것들이 있다
  - 범주화(data suppression)
    - 데이터값을 범주화하여 명확한 값을 대체
    - 예) 김치국, 38세, 수원 거주 -> 김치국, 30대, 경기도 거주
  - 데이터 마스킹(data masking)
    - 개인 식별에 유의한 변수값을 보이지 않도록 처리
    - 예) 김치국, 38세, 수원 거주 -> 김**, 38세, 수원 거주

2. 안전성 확보 조치 기준

개요

- AI 개발자 및 서비스 제공자는 정당한 이익과 정보주체 권리 사이의 명백한 우선관계를 확인하기 어려운 경우, 정보주체 권리에 대한 제약 또는 침해를 예방·방지하기 위한 안전성 확보 조치를 충분히 시행하는 것이 바람직함

학습데이터 수집 출처 검증·관리

- 사례 : 데이터 출처 검증을 위한 고려사항

1. 불법 복제물, 아동 성착취물 등 위법한 데이터가 거래되거나 거래될 가능성이 높은 도메인 (예: 딥웹, 다크웹)으로부터 학습데이터 수집 금지
2. 개인정보가 집적되어 있을 개연성이 높은 웹사이트(예: 개인정보 색인·거래 사이트) 배제
3. 로봇배제표준(robots.txt) 준수
4. 저작권, 디자인권 등 지식재산권 존중

2. 안전성 확보 조치 기준

학습데이터 수집 출처 검증·관리

● 사례 : 개인 식별자 삭제 또는 비식별화 사례

발화데이터 비식별화 예시

[혈액형]
나의 혈액형은 [BLOOD_TYPE_1]. 어떤 혈액형과의 성향이 가장 잘 맞을까?
[병명]
나 [CONDITION_1] 어제 쉬었어.
[CONDITION_2] 좋은 음식이 뭐가 있니?
[복용약/량]
나 어제 [DRUG_1] 먹었는데, 효과 좋더라. 너도 이걸로 먹어. 하루에 [DOSE_
[향정신성 의약품]
나 아는 형이 [DRUG_2] 줘서 먹었는데, 쩔더라
[외상/상처]
나 [CONDITION_3] 있는데, 어떻게 극복할 수 있을까?
[의료 행위]
어제 [MEDICAL_PROCESS_1], [CONDITION_4]이 심하다는 데, 살 빼려면 어
어제 [MEDICAL_PROCESS_2], 무릎 연골이 부숴져서 나 이제 못 걸을 수 있다
어제 [OCCUPATION_1] 쌤이 [CONDITION_5] [MEDICAL_PROCESS_3]? 같은
[전문적/통계적 지식]
어제 [OCCUPATION_1] 쌤한테 들었는데, [CONDITION_6] 치료율이 [STATIS
[은행_계좌]
우리 은행 [BANK_ACCOUNT_1] [MONEY_1] 보내줘
하나 [CREDIT_CARD_1] 주인 이름이 어떻게 되니?
[신용카드]
[ORGANIZATION_1] [CREDIT_CARD_2] [CREDIT_CARD_EXPIRATION_1] [
[금융_거래정보]

■고유식별정보
- 주민등록번호
- 운전면허번호
- 여권번호

■민감정보
- 종교
- 정치이념
- 노동조합명
- 질병명
- 범죄 경력 자료

■기타개인정보
- 계좌번호
- 신용카드정보 등

발화데이터 비식별화 예시(Wrtn)　　비식별화 정보 유형 예시

2. 안전성 확보 조치 기준

미세조정을 통한 안전장치 추가

- 학습데이터에는 편향적이거나 부정확한 정보, 민감한 사적정보가 포함될 수 있어 사전 정제 과정이 수반되는 경우가 많으나, 이로써 모든 위험이 예방되는 것은 아니기 때문에 미세조정(Supervised Fine-Tuning, SFT), 사람 피드백 기반 강화 학습 (Reinforcement Learning with Human Feedback, RLHF) 등의 미세조정 기법 적용을 고려할 수 있음

❖ 최근 RLHF에 소요되는 막대한 비용(사람 레이블러 동원에 필요한 비용)과 사람의 주관적 편향성, 기술적 복잡성 등에 대한 한계를 보완하기 위하여 RLHF를 대체하는 방법론(예: Direct Preference Optimization, DPO)등이 꾸준히 연구 중으로, 향후 이러한 기술적 발전을 고려하여 안전장치를 확보하는 것이 바람직함

2. 안전성 확보 조치 기준

미세조정 종류

- 파라미터 효율 미세조정(Parameter Efficient Fine-Tuning(PEFT))
  - 사전학습된 모델 파라미터(매개변수)를 동결하고 소수의 파라미터를 의도된 용도에 맞게 미세조정하는 것으로 학습 비용과 시간을 최소화하는 방법
- 지도학습 기반 미세조정(Supervised Fine-Tuning(SFT))
  - 비지도학습으로 만들어진 생성AI를 지도학습적으로 미세조정하는 과정으로, 바람직한 답변을 생성하도록 미리 정제되거나 레이블링된 데이터를 추가 학습
  - ※ (예) 개인의 사생활을 묻는 프롬프트에 대하여 답변을 거부하는 내용의 답안을 학습시킴

2. 안전성 확보 조치 기준

미세조정 종류

- 사람 피드백 기반 강화학습(Reinforcement Learning with Human Feedback(RLHF))
  - 보상모델 생성(Reward Model Creation) : AI 모델이 생성한 출력물에 사람(라벨러)이 점수 또는 순위를 부여하고, 이를 토대로 보상모델을 훈련
  - ※ (예) 개인의 사생활을 묻는 프롬프트에 대하여 사생활이 포함된 답변에는 (-1)의 보상을, 회피하는 답변에는 (+1)의 보상을 제공
  - 정책 최적화(Policy Optimization): 보상모델을 사용하여 AI 모델의 정책을 최적화하는 단계로, 주로 정책 그라디언트 강화학습 알고리즘인 PPO(Proximal Policy Optimization)을 활용하여 미세조정

2. 안전성 확보 조치 기준

미세조정을 통한 안전장치 추가

- 다양한 미세조정 방식 비교

다양한 미세조정 기법	파라미터 효율 미세조정 (PEFT)	지도학습 기반 미세조정 (SFT)	사람 피드백 기반 강화학습 (RLHF)
	Base LLM Tunable 미세조정 답변생성	Base LLM Tunable 미세조정 답변생성	SFT 등 LLM Tunable 미세조정 답변생성
방법	소수의 파라미터 조정(~0.01%)	이상적 답변 생성을 위한 추가학습	보상모델 생성 및 정책 최적화
학습데이터	X백개 이상	X만개 ~ XX만개	X만개 ~ XX만개
학습 비용	비교적 저렴	비교적 비쌈	비교적 비쌈
학습 시간	Minutes	Days	Days

3. Metric Evaluation

모델 성능 평가(Metric Evaluation)?

- 머신러닝과 데이터 분석에서 **모델 성능 평가(Metric Evaluation)**는 결과의 신뢰성과 품질을 판단하는 핵심 과정.
- 그중 Accuracy는 가장 기본적이고 직관적인 지표이며, 다양한 성능 지표와 함께 사용.

성능 측정 지표 (Performance Metrics)란?

- 성능 지표는 모델이 예측을 얼마나 잘 수행했는지를 수치적으로 표현한 값

3. Metric Evaluation

성능 측정 개요

- 객관적인 성능 측정의 중요성
  - 모델 선택할 때 중요
  - 현장 설치 여부 결정할 때 중요
- 일반화generalization 능력
  - 학습에 사용하지 않았던 새로운 데이터에 대한 성능
  - 가장 확실한 방법은 실제 현장에 설치하고 성능 측정 → 비용 때문에 실제 적용 어려움
  - 주어진 데이터를 분할하여 사용하는 지혜 필요
  - O/X의 문제가 아닌, 얼마나 일치하는가?

3. Metric Evaluation

성능 측정 개요

- 여러가지 평가 방법들
  - Confusion Matrix 기반
  - Accuracy
  - Precision
  - Recall
  - F1 score
  - AUC (Area Under the Curve) & ROC (Receiver Operating Characteristic)

3. Metric Evaluation

혼동 행렬과 성능 측정 기준

- 혼동 행렬(Confusion Matrix)

  - 부류 별로 옳은 분류와 틀린 분류의 개수를 기록한 행렬
  - 이진 분류에서 긍정positive과 부정negative
  - 검출하고자 하는 것이 긍정(환자가 긍정이고 정상인이 부정, 불량품이 긍정이고 정상이 부정)
  - 참 긍정(TP), 거짓 부정(FN), 거짓 긍정(FP), 참 부정(TN)의 네 경우

	실제 Positive	실제 Negative
예측 Positive	**TP** (True Positive)	**FP** (False Positive)
예측 Negative	**FN** (False Negative)	**TN** (True Negative)

3. Metric Evaluation

혼동 행렬과 성능 측정 기준

- 위의 혼동 행렬로부터 계산될 수 있는 주요 평가 지표(Metric)는 대표적으로 정확도 (accuracy), 오차 비율(error rate) = 1-정확도, 민감도(sensitivity 혹은 재현율 : Recall, Hit Ratio, TP Rate 등으로도 부름), 특이도(specificity), 거짓 긍정률(FP Rate), 정밀도 (precision) 등이 있으며, 이 중에서 정확도, 민감도, 정밀도 등은 특히 많이 사용되는 지표이다.
- 또한, 민감도와 정밀도를 조합한 F-Measure(혹은 F1-Score) 및 분석 모델의 예측값 과 실제값이 정확히 일치하는 정도를 계량화한 카파 통계(Kappa Statistic) 등이 정의되어 있다.

3. Metric Evaluation

주요 성능 지표 정리

지표	수식	설명
정확도 (Accuracy)	(TP + TN) / (TP + TN + FP + FN)	전체 샘플 중 정답 비율
정밀도 (Precision, PPV)	TP / (TP + FP)	Positive 예측 중 실제 Positive 비율
재현율 / 민감도 (Recall, Sensitivity, TPR)	TP / (TP + FN)	실제 Positive 중 맞춘 비율
특이도 (Specificity, TNR)	TN / (TN + FP)	실제 Negative 중 맞춘 비율
NPV (Negative Predictive Value)	TN / (TN + FN)	Negative 예측 중 실제 Negative 비율
F1-Score (F-Measure)	2 × (Precision × Recall) / (Precision + Recall)	정밀도와 재현율의 조화 평균
Fall-out (FPR, 위양성률)	FP / (FP + TN)	실제 Negative 중 잘못 Positive로 예측한 비율
Balanced Accuracy	(TPR + TNR) / 2	민감도와 특이도의 평균

3. Metric Evaluation

학습 데이터셋, 테스트 데이터셋

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df_train = pd.read_csv("datasets/titanic_train.csv")
df_test = pd.read_csv("datasets/titanic_test.csv")
df_train.head(5)

from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer

# 특성과 타깃 분리
X_train = df_train.drop(columns=['survived'])
y_train = df_train['survived']
X_test = df_test.drop(columns=['survived'])
y_test = df_test['survived’]

# 문자열 데이터를 원-핫 인코딩하기 위한 열 선택
categorical_features = X_train.select_dtypes(include=['object']).columns
numeric_features = X_train.select_dtypes(exclude=['object']).columns
```

3. Metric Evaluation

모델링

```
# 전처리 파이프라인 생성
preprocessor = ColumnTransformer(
    transformers=[
        ('num', Pipeline([
            ('imputer', SimpleImputer(strategy='mean')),
            ('scaler', StandardScaler())
        ]), numeric_features),
        ('cat', Pipeline([
            ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
            ('onehot', OneHotEncoder(handle_unknown='ignore'))
        ]), categorical_features)
    ])
# Logistic Regression 분류 모델링

from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_curve, roc_auc_score
import matplotlib.pyplot as plt


# 로지스틱 회귀 모델을 포함한 파이프라인 생성
lr = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', LogisticRegression(random_state=0))
])
```

3. Metric Evaluation

모델 학습

```
# 모델 학습
lr.fit(X_train, y_train)

# 테스트 데이터셋에 대한 예측
y_pred = lr.predict(X_test)
y_pred_probability = lr.predict_proba(X_test)[:, 1]
```

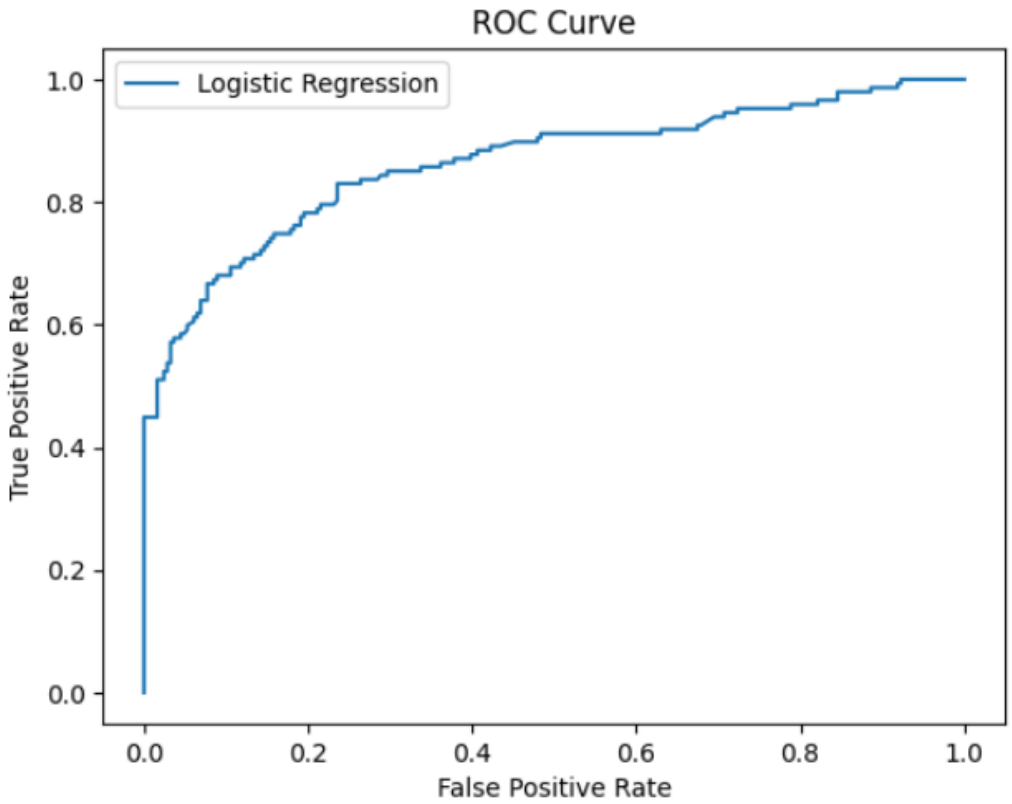
분류 모델 평가

```
# 평가 지표 출력
print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
print(f'Precision: {precision_score(y_test, y_pred)}')
print(f'Recall: {recall_score(y_test, y_pred)}')
print(f'F1 Score: {f1_score(y_test, y_pred)}')
print(f'ROC AUC Score: {roc_auc_score(y_test, y_pred_probability)}')
```

3. Metric Evaluation

ROC Curve 그리기

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_probability)
plt.plot(fpr, tpr, label='Logistic Regression')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```



3. Metric Evaluation

분류 모델 평가

```
from sklearn.metrics import confusion_matrix

# Confusion Matrix를 출력합니다.
confmat = confusion_matrix(y_true=y_test, y_pred=y_pred)
print(confmat)
```

```
[[216  30]
 [ 44 103]]
```

accuracy: 0.811704834605598
Precision : 0.756
Recall : 0.673
F1 : 0.712

(216+103) / (216+103+44+30)

실제값(Reference)

예측값(Prediction)	Y	N
Y	True Positive(TP)	False Positive(FP)
N	False Negative(FN)	True Negative(TN)

Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	전체 예측에서(예측이 Y이든 N이든 무관하게) 옳은 예측의 비율

2. 실습

실습21 : 데이터 변환-1

문제
[재범주화데이터.xlsx]파일 첨부
칼럼 중에서 값의 길이가 10이 넘는 칼럼을 알려 주세요.

2. 실습

실습22 : 데이터 변환-2

문제
[날짜데이터_변환전.xlsx]파일첨부
첨부한 파일의 데이터를 보여 주세요.

2. 실습

실습23 : 데이터 취합하기

문제
[날짜데이터_변환완료.xlsx]파일 첨부
이 데이터 파일의 shape을 알려 주세요.

THANK YOU.

앞으로의 엔지니어는 단순한 '코더'나 '기계 조작자'가 아니라 뇌-기계 인터페이스를 통해 지식과 능력을 즉각 확장하는 존재(뉴로-인터페이스: Neuro Interface)가 될 수 있습니다.

🎯 목표 달성을 위한 여정이 시작됩니다.
✨ 궁금한 점이 있으시면 언제든 문의해주세요!
🚀 함께 코더와 프롬프트 전문가로 성장해 나갑시다!