programmers

2. 데이터 EDA와 머신 러닝 소개

머신 러닝 모델링 시작인 데이터 분석에 대해 배우고
일반적인 머신 러닝 방식에 대해 학습하자

한기용

keeyonghan@hotmail.com

programmers

목차

programmers

데이터 EDA란?

EDA(Exploratory Data Analysis)에 대해 알아보자

EDA를 하는 이유 (1)

- 효과적인 데이터 분석과 모델링을 위한 기초 마련
- 데이터 품질 확인: Garbage In Garbage Out
- 데이터 특성 (패턴) 확인

EDA는 데이터 분석이나 모델링 전에 꼭 수행해야하는 작업!!
특히 데이터 품질이란 측면에서는 항상 확인이 필요하다는 점 명심!!

EDA를 하는 이유 (2)

- 현업에서 깨끗한 데이터란 존재하지 않음
  - 항상 데이터를 믿을 수 있는지 의심할 것! -> **의(疑)데이터증**
  - 실제 레코드를 몇 개 살펴보는 것 만한 것이 없음 -> **노가다**
- 데이터 일을 한다면 항상 데이터의 품질을 의심하고 체크하는 버릇이 필요
  - 중복 레코드 체크하기 (duplicates)
  - 최근 데이터의 존재 여부 체크하기 (freshness)
  - Primary key uniqueness가 지켜지는지 체크하기
  - 값이 비어있는 컬럼들이 있는지 체크하기
  - …

데이터 셋 이해를 위한 일반적인 방법들

- 기술 통계 분석
- 결측치 탐지 및 처리
- 이상치 탐지 및 처리
- 데이터 시각화
- 상관 관계 분석
- (고급) 피처 엔지니어링

programmers

데이터 EDA 기법 소개와 실습

EDA(Exploratory Data Analysis)에 대해 알아보자

데이터 전처리 기법 - 기술 통계 분석

- 숫자 변수와 카테고리 변수 파악
- 숫자 변수의 경우 값 범위 파악
- 카테고리 변수의 경우에는 카테고리 수 파악
- 카테고리의 경우 머신 러닝 모델을 만들때 인코딩 방법 결정

데이터 전처리 기법 - 결측치 탐지 및 처리

- 비어있는 값이 있는 필드를 가진 레코드들을 찾기
- 해당 필드들을 어떻게 할지 결정이 필요
  - 그런 레코드를 무시하는 것도 하나의 결정
  - 필드를 채우기로 했다면
    - 숫자 필드의 경우 평균값, 최소값, 최대값, 가장 흔한 값등을 기본값으로 사용 가능
    - 카테고리 필드의 경우 가장 흔한 카테고리 값을 기본값으로 사용 가능
- Pandas의 경우에는 isnull().sum() 함수 사용

데이터 전처리 기법 - 이상치 탐지 및 처리

- 숫자 필드의 경우 아주 크거나 작은 값을 갖는 소수의 레코드가 있다면?
- 여러 가지 처리 방법이 존재
  - 해당 레코드들을 전체적으로 다 무시
  - 그 숫자 필드의 값을 전체 평균값이나 최소값이나 최대값등으로 교체
    - 예를 age 필드의 값이 1000이라면 전체 평균을 대신 사용
  - 그 숫자 필드의 값을 다른 값으로 전환
    - 예를 들어 log를 적용
    - 예를 들어 binning을 적용
  - 이상치에 강한 ML 모델링 방식을 사용
    - 예를 들면 Decision Tree

데이터 전처리 기법 - Primary Key Uniqueness 체크

- 데이터에 Primary Key가 존재하는 경우라면?
- Primary Key의 값이 유일함을 꼭 검증하는 것이 좋음

데이터 전처리 기법 - 최신성 체크

- 데이터가 최신 데이터이어야 한다면?
- 데이터에 존재하는 타임스탬프 필드를 기준으로 최신 데이터가 있는지 꼭 확인

데이터 전처리 기법 - 레이블 (타켓) 체크

- 예측 대상이 되는 필드의 값이 어떻게 분포되어있는지 확인
- 분류 모델인데 레이블 값 분포가 한쪽으로 치우쳤다면 평가 지표를 F1으로 사용
  - 또한 부족한 예들을 찾아서 훈련 데이터에 추가하는 노력이 필요함
  - 이는 이미지, 오디오 등에서는 조금 더 쉽고 다양한 파이썬 모듈들이 존재함
    - 다른 데이터셋에서는 SMOTE(Synthetic Minority Over-sampling Technique) 사용 가능

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

데이터 전처리 기법 - 변수간 상관 관계 검사

- 특히 레이블과 상관관계를 보여주는 변수들이 무엇인지 검사
  - Correlation 보기
- Python Pandas라면 corr() 함수 사용

데이터 전처리 기법 - 피처 엔지니어링

- 앞서 이야기한 모든 기법들이 피처 엔지니어링
- 조금더 고급 기법으로는
  - 필드의 수가 너무 커지면 PCA 등을 통해 필드 수를 줄여보는 것
  - 앞서 언급한 이름 필드에서 성별등을 추출해보는 것
  - 특히 Regression의 경우 레이블 필드와 약한 상관관계를 갖고 있는 필드들을 곱하거나 더해서 새로운 필드를 만드는 것 등등

판다스 사용 전처리

- CSV 파일 읽기: read_csv를 사용하며 URL을 인자로 사용하는 것이 가능
- 레코드 일부 확인해보기: head 혹은 tail
- 전체적인 데이터 특징 살펴보기: unique, describe와 info
  - 이를 통해 데이터의 특징을 쉽게 살펴볼 수 있다
- 데이터 필터링: where, dropna, drop
- 데이터 시각화: hist, boxplot, plot

실습

1. 타이타닉 데이터셋을 바탕으로 진행
2. Kaggle 노트북 하나를 가지고 실습 진행
    a. https://www.kaggle.com/code/mjamilmoughal/eda-of-titanic-dataset-with-python-analysis

programmers

머신 러닝이란?

머신 러닝이 무엇이고 어떤 종류들이 있는지 알아보도록 하자

머신 러닝(Machine Learning)의 정의

- Machine Learning:
  - ‘A field of study that gives computers the ability to learn without being explicitly programmed’ (Arthur Samuel)
- “배움이 가능한 기계의 개발”
  - 결국 데이터의 패턴을 보고 흉내내는 방식 (imitation)
  - 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야
  - 딥 러닝(신경망의 다른 이름)은 이 중의 일부
    - 비전, 자연언어처리 (텍스트/오디오)등에 적용되고 있음
  - AI는 머신러닝을 포괄하는 개념

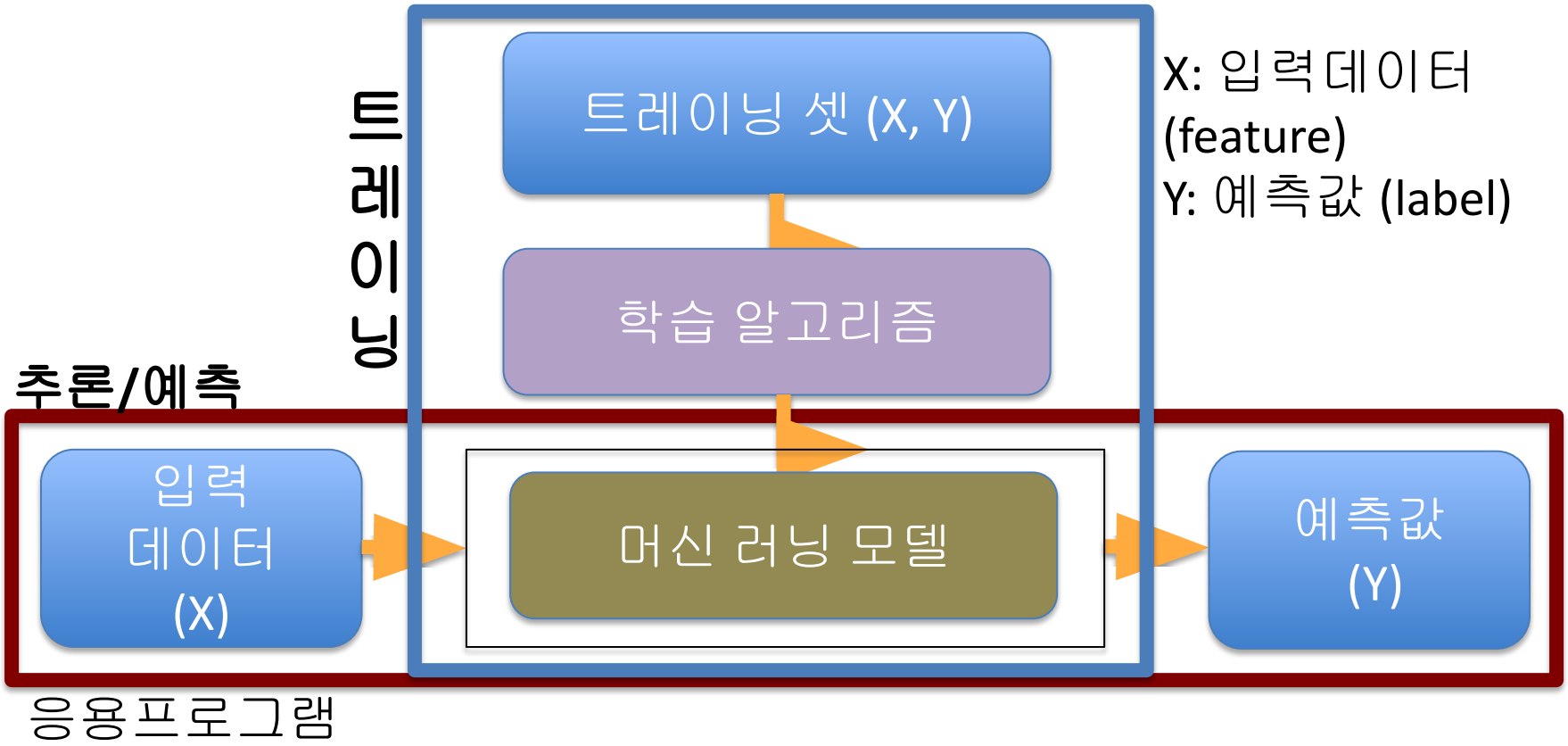
머신 러닝 모델이란?

- 머신 러닝을 통해서 최종적으로 만드는 것이 머신 러닝 모델
  - 특정 방식의 예측을 해주는 블랙박스
    - 선택한 머신 러닝 알고리즘에 따라 내부가 달라짐
    - 디버깅은 쉽지 않음
  - 입력 데이터를 주면 그를 기반으로 예측
    - 정확히 이야기하자면 Supervised ML (지도기계학습)
- 모델 트레이닝/빌딩
  - 머신 러닝 모델을 만드는 것을 지칭

머신 러닝의 종류

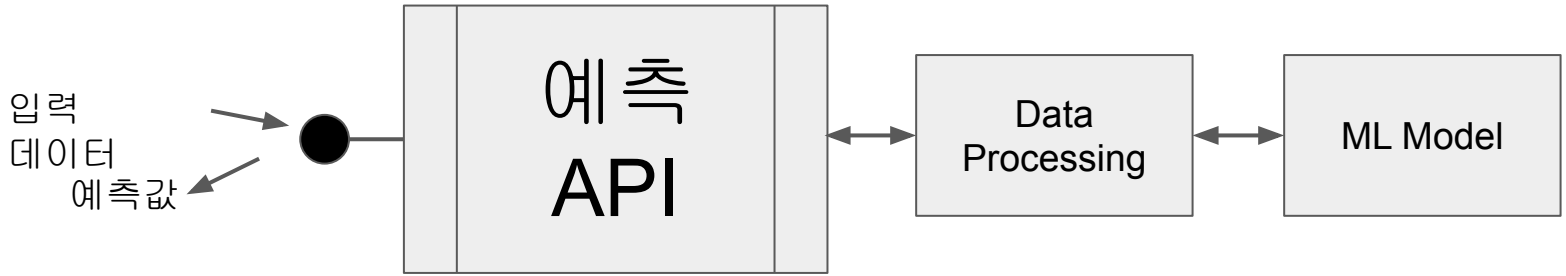
- 지도 기계 학습 (Supervised Machine Learning)
  - 명시적 예제 (트레이닝셋)을 통해 학습: 정답이 존재
  - 크게 두 종류가 존재
    - 분류 지도 학습 (Classification): 이진 분류(Binary)와 다중 분류 (Multi-class)
    - 회귀 지도 학습 (Regression)
- 비지도 기계 학습 (Unsupervised Machine Learning)
  - 클러스터링 혹은 뉴스 그룹핑처럼 주어진 데이터를 몇 개의 그룹으로 분리
  - GPT 같은 언어 모델의 훈련도 여기에 속함 (Semi-Supervised Machine Learning)
- 강화 학습 (Reinforcement Learning)
  - 시행착오를 통해 최적의 결정을 학습하는 기계학습 방법
  - 알파고 혹은 자율주행

지도 기계 학습


트레이닝
트레이닝 셋 (X, Y)
학습 알고리즘
X: 입력데이터 (feature)
Y: 예측값 (label)
추론/예측
입력 데이터 (X)
머신 러닝 모델
예측값 (Y)
응용프로그램

API란?

- Application Programming Interface의 약자
  - 인터페이스란 무엇인가와 통신을 하기 위한 방법을 의미
  - 프로그램을 작성하기 위해 사용하는 인터페이스
- 다양한 종류의 API가 존재
  - 예를 들어 파이썬 모듈은 각기 제공해주는 기능에 따라 다양한 함수들을 제공
    - 이 것도 API라고 부를 수 있음
  - 보통 API라고 하면 웹상의 다른 서버에 존재하는 특정기능을 사용가능하게 해주는 인터페이스를 지칭
- ML 문맥에서 API는 결국 모델을 통해 예측하는 것을 의미



지도 기계 학습 예제: 타이타닉 승객 생존 여부 예측

- 이진 분류 문제 (Binary Classification)
- 탑승 승객별로 승객 정보와 최종 생존 여부가 트레이닝셋으로 제공됨
  - 최종 생존 여부처럼 모델이 예측해야하는 필드를 레이블/타켓이라고 부름
  - 기존 필드로부터 새로운 필드를 뽑아내는 것이 일반적: Feature Engineering

survived,pclass,name,sex,age,sibsp,parch,ticket,fare,cabin,embarked
1,2,"Hewlett, Mrs. (Mary D Kingcome) ",female,55,0,0,248706,16,,S
0,3,"Rice, Master. Eugene",male,2,4,1,382652,29.125,,Q
…

지도 학습 예: 스팸 웹 페이지 분류기 (Classification)

"클라우드 컴퓨팅"(Cloud Computing)이란 집적·공유된 정보통신기기, 정보통신설비, 소프트웨어 등 정보통신자원을 이용자의 요구나 수요 변화에 따라 정보통신망을 통하여 신축적으로 이용할 수 있도록 하는 정보처리체계를 말한다(클라우드컴퓨팅 발전 및 이용자 보호에 관한 법률 제2조 제1호).

A

클라우드 컴퓨팅, 온라인 도박, 간편즉시 대출, 정보통신기기, 신용불량 대출, 온라인 카지노, 교통사고 상해 변호사, 주식투자, 부동산투자, 중고차매매, 해외여행, 저가항공편, ..

B

어느 쪽이 정상 웹 페이지일까?
어떤 특징(feature)을 뽑아내면 정상인지 스팸인지 결정하는데 도움이 될까?

지도 학습 예: (Regression)

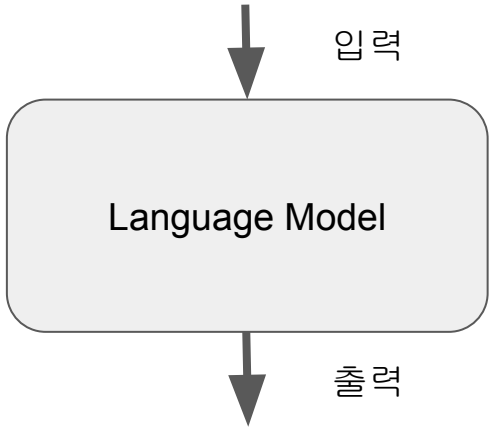
- 회귀는 연속적인 숫자를 예측하는 데 사용되는 머신 러닝 모델
- 주택 가격 예측이 Regression 모델링의 예
  - 집 평수와 판매 가격과 우편번호로 구성된 주택 정보 데이터셋 존재
  - 주택 평수와 우편번호를 바탕으로 판매 가격을 예측: Regression 모델링

주택 평수	우편 번호	판매 가격
38	01000	120,000,000
28	01001	75,000,000
32	01002	99,000,000

비지도 학습 예: Language Model

- 문장의 일부를 보고 비어있는 단어를 확률적으로 맞추는 모델
- 훈련은 위키피디아에 있는 자연스러운 문장들을 대상으로 진행

Seoul is the capital of ()



Seoul is the capital of (Korea)
(South Korea)
(Republic of Korea)

(OpenAI) (transitioned) (from)
(non-profit) (to) (for-profit)

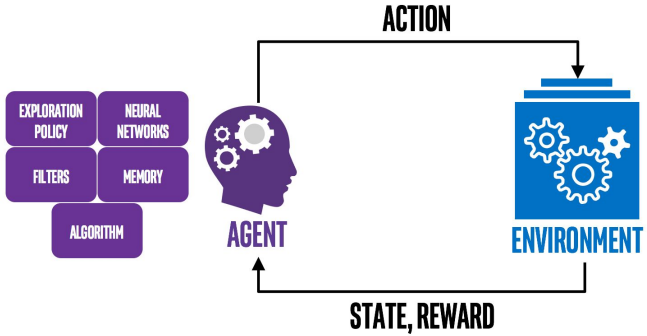
[“OpenAI transitioned from”,
“non-profit”]
[“transitioned from non-profit”, “to”]
[“from non-profit to”, “for-profit”]

위의 경우 context window가 4가 됨:

- 3개의 토큰을 보고 1개의 토큰 예측을 훈련
- Context window의 크기가 결국 모델의 메모리를 결정

강화학습이란?

- 시행착오를 통해 최적의 결정을 학습하는 기계학습 방법
- 에이전트(강화학습 모델)는 환경과 상호작용하며 최대 보상 전략을 학습
  - 에이전트는 행동에 대한 보상을 받아 시간이 지남에 따라 최대의 보상을 얻을 수 있는 전략을 학습
- 중요 개념
  - 에이전트(Agent)와 환경(Environment)
  - 상태(State)와 행동(Action)
  - 보상(Reward)과 처벌(Punishment)
- 강화학습의 예:
  - 게임 플레이 (ex: 바둑, 체스, ...)
  - 자율 주행
  - 로봇 (ex: 자동차 조립)



medium.com

programmers

머신 러닝 관련 개념

머신 러닝 모델링 관련 개념과 용어들에 대해 알아보도록 하자

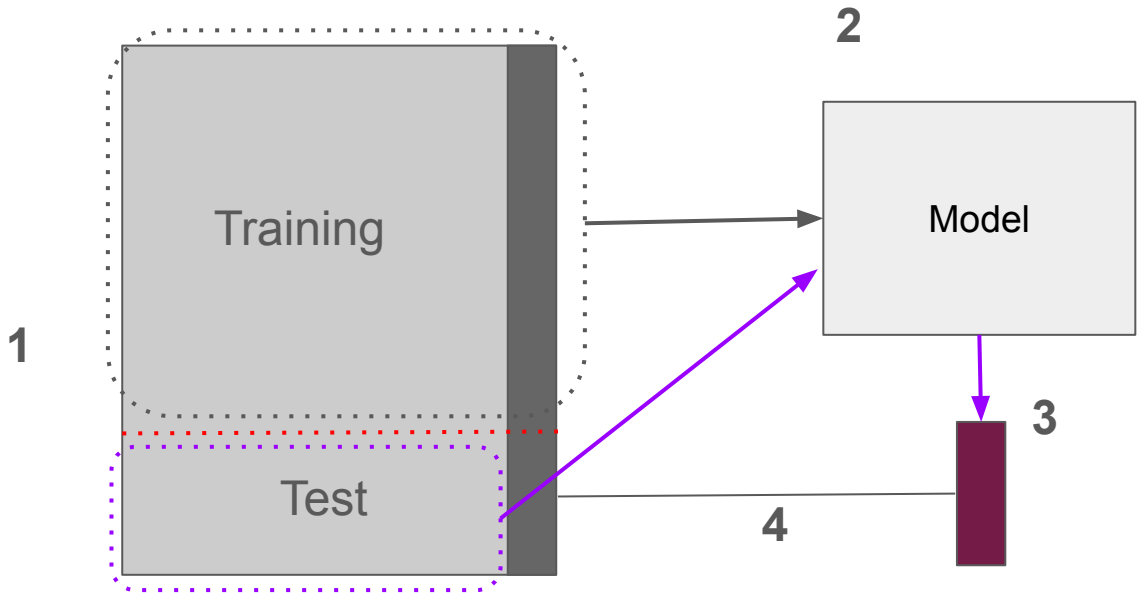
과적합(Overfitting)이란?

- 모델이 트레이닝 셋 데이터에 너무 최적화된 경우
  - 새로운 데이터에서 성능이 제대로 나오지 않는 경우
- 이를 완화하는 방법
  - 데이터 셋 수집에 바이어스가 있는지 확인
  - 모델 성능 평가를 hold-out 테스트보다는 cross-validation 테스트를 사용 (뒤에서 더 언급)
  - 정규화를 적용함 (뒤에서 더 언급)

Hold Out이란 ?

- Overfit을 방지하기 위한 가장 간단한 방법
  - 트레이닝 셋을 트레이닝을 위한 용도와 테스트를 위한 용도로 분리
  - 보통 75%:25%나 80%:20%를 사용
- 모델 빌드시 트레이닝 데이터를 사용
  - 그 모델의 성능은 테스트 데이터로 측정

Hold Out - 시각화

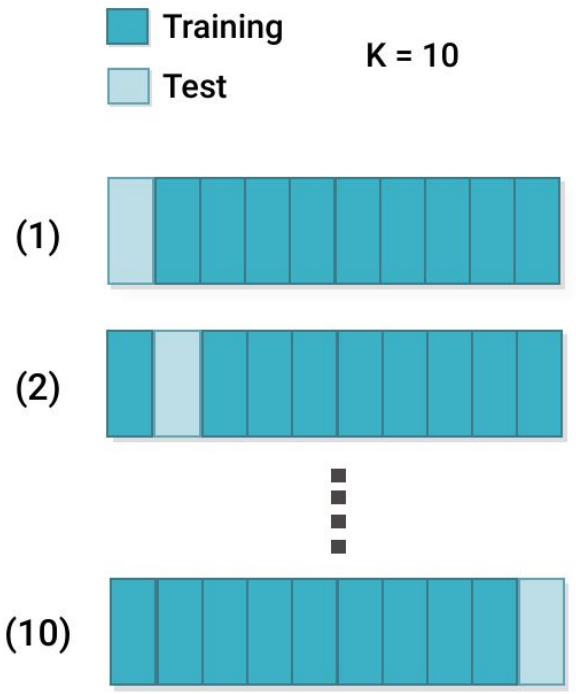


Cross-Validation이란 ?

- K-Fold의 다른 이름
  - Hold-out 방식보다 우월
- 먼저 데이터를 K개로 나눔
  - 대개 5개/10개/20개. 각각을 Fold라고 부름.
- 이 K개의 폴드에 대해서 아래를 반복
  - 해당 폴드를 제외한 나머지 데이터로 모델을 빌드
  - 해당 폴드 데이터로 검증
- 결국 K번의 모델 빌딩과 성능측정이 수행됨
  - 모든 폴드 처리후 K개의 성능 결과 평균을 모델의 전체 성능으로 간주

Cross-Validation: 시각화

- 기본적으로 hold-out을 K번 반복하는 방식



표준화(Normalization) #1

- 트레이닝셋이 특정 방향으로 바이어스되지 않도록 보정해주는 것
  - 특정 필드의 값이 더 커서 그게 더 큰 영향을 끼치는 것을 방지
- 예) 정자세로만 있는 데이터셋으로 학습된 모델이 약간 뒤틀린 자세로 있는 사진도 인식하게하는 용도

표준화(Normalization) #2

- 트레이닝셋에 존재하는 feature들의 값을 특정 범위로 제약을 주어 모델의 성능이 트레이닝셋에 따라 달라지는 것을 방지
  - 보통 모든 feature들의 값을 동일한 범위에 들어가도록 하는 전처리 기법
  - 예) 최대/최소값이 각각 1과 -1이 되도록 표준화
  - 예) 최대/최소값이 각각 1과 0이 되도록 표준화
- Deep Learning에서는 “Batch Normalization”이라는 것이 존재

비용 함수 (Cost Function, Loss Function) #1

- 모델의 예측 정확도를 측정하는 목적으로 사용되는 함수
- 이 함수가 최소값일때의 모델이 바로 최적의 모델

실제 판매 가격	예측 판매 가격
120,000,000	110,000,000
75,000,000	81,000,000
99,000,000	98,000,000

생존 여부	생존 예측
1	1
0	1
1	1
1	1

비용 함수 (Cost Function, Loss Function) #2

- 모델의 예측 정확도를 측정하는 목적으로 사용되는 함수
- 이 함수가 최소값일때의 모델이 바로 최적의 모델
- 비용 함수의 종류
  - Absolute loss (Least Absolute Deviation, L1 norm)
  - Square loss (Least Square Error, L2 norm)
  - Hinge loss
  - Logistic loss
  - Cross entropy loss
  - RMSE (Root Mean Squared Error)
  - Logarithmic loss (RMSLE, Root Mean Squared Logarithmic Error)

정규화 (Regularization)

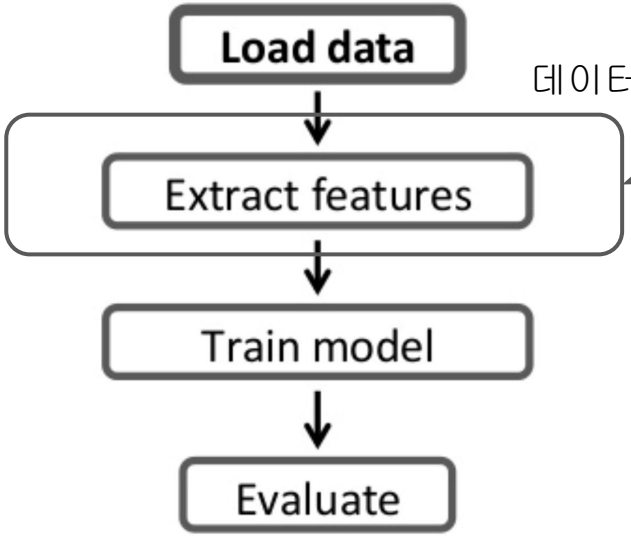
- Overfitting을 막기 위한 사용하는 방법 중의 하나
  - 과적합은 모델이 훈련 데이터에 최적화되어 새로운 데이터 성능이 저하되는 것
  - 예) 흰색 강아지만 있는 데이터셋으로 트레이닝된 모델에서 검정색 강아지도 인식하지 못함
- 정규화는 손실 함수에 추가 정보(또는 페널티) 추가로 과적합을 방지하는 기술
  - 일부 피처의 가중치를 0으로 만들거나 아주 작게 만드는 것
- 정규화의 가장 일반적인 두 가지 형태는 L1 및 L2 정규화
  - 정규화 없는 ML 방식: Linear Regression
  - L1 정규화: 일부 덜 중요한 피처의 가중치를 0으로 만듬
    - Lasso Regression
  - L2 정규화: 일부 덜 중요한 피처의 가중치를 아주 작게 만듬
    - Ridge Regression

programmers

모델 추론 과정 소개

모델을 만들고 난 뒤 사용할 때 어떤 과정들이 필요한지 알아보자

예측 데이터를 실제 훈련에 사용되는 데이터로 전환



모델 훈련 과정

데이터 처리 과정

- 훈련용 데이터가 있는 그대로 모델 훈련에 사용되는 것이 아니라 숫자가 아닌 값들은 무시되거나 숫자로 변환되고 숫자들은 표준화 (0과 1사이) 등으로 한번 변환됨

- 이 변환을 모델 API를 호출하는 사람이 직접 해야한다면?
- 이 변환은 API 내에서 수행해주는 것이 여러모로 좋음!

타이타닉 승객 생존 예측에서의 예

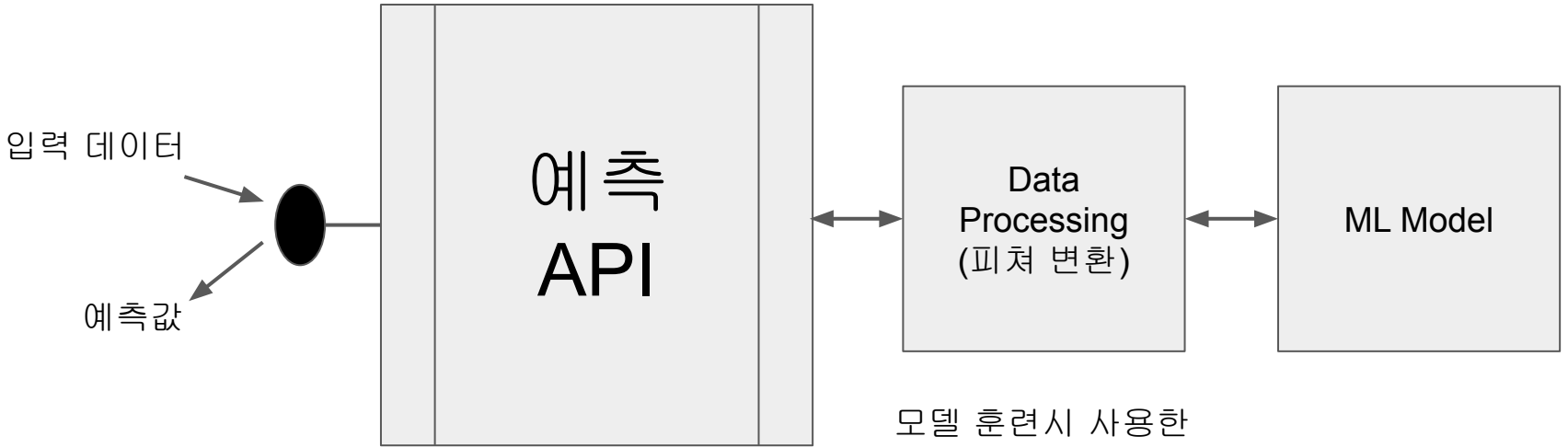
- 원 입력레코드

PassengerId,Pclass,Name,Gender,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
4,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S

- 실제 모델로 들어가는 입력
  - 텍스트 필드들은 모두 숫자로 변환
    - Gender 필드의 경우: female -> 0, male -> 1
  - 숫자 필드들의 경우 표준화가 필요 (-1 ~ 1, 0 ~ 1)

Pclass,Name,Gender,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked

일반적인 예측 API의 모양



모델 훈련시 사용한 것과 동일한 데이터 처리과정을 통하는 것이 중요

모델 추론 과정 모니터링이 중요해짐

- 모델 빌딩을 하는 사람과 추론 운영을 하는 사람이 보통 다름
  - 여기서 다양한 문제들이 발생함
  - 그래서 만들어진 팀이 MLOps
    - 이걸 LLM과 관련해서 전문화한 LLMOps 등 다양한 팀이 만들어지고 있음
- 데이터의 패턴이 달라지면서 모델의 성능이 떨어지기 시작함
  - Data Drift라고 부름
  - 모델의 중요 feature의 값 분포 모니터링과 모델 관련 중요 지표 모니터링이 꼭 필수

programmers

Q & A

오늘 강의에 대해서 궁금한 부분이 있으면 알려주세요!