

<Hands-On Machine Learning.>

Part 1. 머신러닝

chapter 1. 한눈에 보는 머신러닝

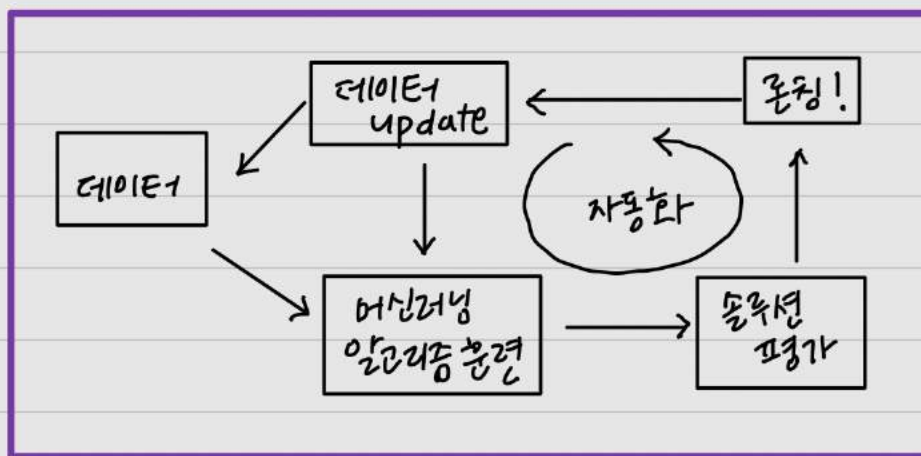
1.1. 머신러닝이란?

어떤 작업 T 에 대해 성능 P (정확도)가 경험 D (훈련데이터)로 인해 성능이 향상되었으면 컴퓨터가 경험 D 로 학습. 즉 머신러닝 함을 의미.

1.2 왜 머신러닝을 사용하는가?

전통적 → 규칙이 점점 늘고 복잡해져 유지보수하기 힘들

머신러닝 → 패턴감지. 자동학습. 정확도 ↑



데이터 마이닝 : 대용량의 data → 패턴발견 (by 머신러닝기술)

1.3 머신러닝 시스템의 종류

1.3.1 지도/비지도 (사람의 감독 여부로 구분)

(1) 지도학습

- 분류
- 회귀: 연속변수 (특성)을 이용해 타겟 수치 예측
- * 레이블의 범주 = 클래스



ex) K-최근접 이웃, 선형 회귀, 로지스틱 회귀, 서포트 벡터 머신 (SVM), 결정트리, & 랜덤포레스트, 신경망

(2) 비지도 학습

→ 레이블이 없는 훈련데이터

- 군집: K-평균, 계층 군집분석, 기댓값 최대화
- 시각화와 차원 축소: 주성분분석(PCA), 커널PCA, 지역적 선형 임베딩, t-SNE
- 이상치 탐지: ↳ 특성추출
- 연관 규칙 학습: 어프라이어리, 이클렛

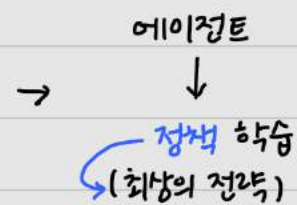
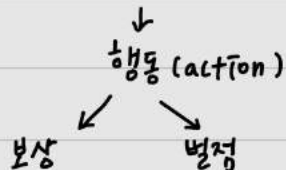
(3) 준지도 학습 (지도+준지도)

→ 레이블 데이터 일부 있음.

- 심층 신뢰 신경망 (DBN) : 비지도 학습인 제한된 볼츠만 머신 (RBM)에 기초

(4) 강화 학습

에이전트 (학습 시스템) ← 환경 (data)



1.3.2 배치 학습과 온라인 학습

(1) 배치 학습 (= 오프라인 학습)

데이터 전체를 이용해 학습시켜야 함. 점진적 불가. (시간, 자원 소요 ↑)

(2) 온라인 학습 (= 미니 배치)

: 작은 묶음단위로 주입하여 학습 (시간, 비용 소요 ↓), 점진적 학습 가능

★ 학습률 [높을 경우: 빠른 적응 but 예전 데이터 금방 잊음
낮을 경우: 느린 적응 but 덜 민감해짐

문제점: 나쁜 데이터 → 시스템 성능 점진적 감소. 모니터링 필요

1.3.3 사례기반 학습과 모델 기반 학습 (how 일반화?)

(1) 사례기반 학습

→ 사례를 기억하고 학습, 유사도 측정 → 새로운 데이터에 일반화

(2) 모델 기반 학습

→ 샘플들의 모델을 만들어 예측에 사용

ex. 선형모델. $\theta_0 + \theta_1 \times \text{인당 GDP}$.

· 모델이 얼마나 좋게 측정하는 '효용함수' (적합도 함수)

· " 나쁜지 측정하는 '비용함수' - 최소화 목적
↳ 훈련시킬 (training)

1.4 머신러닝의 주요 도전과제

→ 문제점 ① 나쁜 알고리즘

② 나쁜 데이터

1.4.1 충분하지 않은 양의 훈련 데이터

1.4.2 대표성이 없는 훈련 데이터

→ 정확한 예측 X. by 샘플링 잡음, 샘플링 편향
↳ (샘플 수 ↓) ↳ 잘못된 표본 추출 방법

1.4.3 낮은 품질의 데이터

예러, 이상치, 잡음 ↑ ⇒ 데이터 정제 필요.

1.4.4 관련 없는 특성

특성공학 (특성선택, 특성추출)
↳ 결합하여 더 유용한 특성 찾기.

1.4.5 훈련데이터 과대적합

과대적합: 모델이 훈련데이터에 잘맞으나 일반성 떨어짐.

· 잡음이 섞인 패턴, 신뢰도 ↓

· 해결방법 ① 파라미터 적은 모델 선택, 특성 수 ↓. 제약걸기

② 훈련데이터 갯수 ↑

③ 잡음의 제거

(= 규제) ... 규제의 양은 '하이퍼 파라미터'가 결정. 훈련전 미리 지정

1.4.6 훈련 데이터 과소적합

과소적합 : 모델이 너무 단순하여 데이터의 내재된 구조 학습하지 못함.

해결방법 ① 파라미터가 더 많은 강력한 모델 선택

② 학습 알고리즘에 더 좋은 특성 제공

③ 모델 제약 감소 (규제 ↓)

1.5 테스트와 검증.

- 훈련세트 / 테스트세트

↳ 일반화 오차 (외부샘플오차)를 얻음

+ 검증세트 (2번째 홀드아웃 세트)

- 교차검증을 이용하여 데이터 배정하지 않기

훈련세트 → 서브셋 (훈련 & 검증) → 일반화 오차 측정

- 평가전에는 더 잘맞는 모델 알수없음.

chapter 2. 머신러닝 프로젝트

2.1 실제 데이터로 작업하기.

2.2 큰 그림 보기

해야 할 일 : 캘리포니아 인구조사 데이터 → 캘리포니아 주택 가격 모델

2.2.1 문제정의

- 비즈니스의 목적이 무엇인지.
- 현재 솔루션은 어떻게 구성되어 있는지.

• 문제정의 : 레이블된 샘플 → 지도 학습

값을 예측 → 회귀 (특성 여러개 : 다변량 회귀)

연속적 X , 레이블 \downarrow → 배치 ← 단변량 회귀.

2.2.2 성능 측정 지표선택

- 회귀 문제의 전형적인 성능 지표 : 평균 제곱근 오차 (RMSE)

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2}$$

오차가 커질수록 커짐.
(유클리디안)

m : 데이터 샘플 수.

h : 예측 함수. (가설)

$x^{(i)}$: i 번째 샘플 전체 특성값의 벡터

$y^{(i)}$: 해당 레이블 (출력값)

- 이상치로 보이는 주격이 많을 경우 : 평균 절대 오차 (MAE)

$$MAE(X, h) = \frac{1}{m} \sum_{i=1}^m |h(x^{(i)}) - y^{(i)}|$$

(맨해튼).

RMSE, MAE 모두 예측값의 벡터와 타겟값의 벡터 사이 거리 재기.

2.2.3 가정 검사.

ex. 카테고리...

2.3 데이터 가져오기.

2.3.3 데이터 구조 보기

· 특성: longitude, latitude, housing-median-age, total-rooms, total-bedrooms, population, household, median-income, median-house-value, ocean-proximity

· 20640 data.

· total-bedrooms 특성 20433, 나머지 null.

· ocean-proximity: 범주형 데이터

· describe(): 숫자형 특성의 요약정보 알려줌



· %matplotlib inline: 주피터 자체의 백엔드를 사용하여 그래프 그림

☆ 히스토그램에서 발견 가능한 사항

① 중간소득: 14 ~ 0.14 (median-income)

② 중간 주택 면적, 중간 주택 소득: 최대, 최소한정
→ 레이블로 사용 (검토 필요).

③ 특성 스케일이 많이 다름

④ 히스토그램  →  변경 필요

2.3.4 테스트 세트 만들기.

→ 데이터 스누핑 편향: 어떤 패턴에 속하 특정 머신러닝 모델 선택.

<무작위 샘플링>

① 테스트 세트 만들고 다음번 실행에 불러들이기

② 난수인덱스 np.random.seed(42)

↓
np.random.permutation()

③ 샘플의 식별자를 이용하기.

* train-test-split 함수

- 난수 초기값 설정가능한 random-state 매개변수

- 둘째 행의 갯수가 같은 여러개의 데이터셋을 인덱스 기반으로 나눌 수 있음.

<계층형 샘플링>

: 계층이라는 동일한 그룹을 나누고 올바른 수의 샘플들을 추출.

→ 계층의 중요도 추정

* 사이킷런의 Stratified ShuffleSplit 사용

: 전체 비율과 테스트 샘플의 비율이 거의 같음을 확인 ↔ 무작위 샘플링

2.4 데이터 이해를 위한 탐색과 시각화.

2.4.1 지리적 데이터 시각화.

① 산점도 ——.plot(kind="scatter", x=" ", y=" ", alpha=0.1)

↓
포인트의 명도, 밀집도 확인 가능.

→ 인구밀도, 지역에 관련이 큼.

2.4.2 상관관계 조사.

• 모든 특성간의 표준상관계수를 corr() 메서드로 계산 가능

↓
 $-1 \leq r \leq 1$
음의 상관관계 양의 상관관계
'0에 가까울수록 관계 ↓'

• scatter-matrix 사용

2.4.3 특성 조합으로 실험

→ 나누고 곱하여 새로운 특성 만들기.

→ 상관관계 계산

2.5 머신러닝 알고리즘을 위한 데이터준비.

- 함수를 자동화해야함

why? ① 손쉽게 변환 반복

② 향후 프로젝트를 위한 변환 라이브러리 구축

③ 새 데이터를 주입하기 전 변환

④ 여러가지 시도 & 조합 판단가능

2.5.1 데이터 정제

· null값에 대한 옵션 - 해당구역제거, 전체특성 삭제, 어떤 값으로 채우기.

→ 사이킷런 Imputer 사용 (누락값처리 → 학습된 중간값으로)

2.5.2 텍스트와 범주형 특성 다루기.

• 범주형 → 숫자형으로 변환 (factorize() 메서드)

문제: 숫자가 가까울 수록 비슷하다고 생각하지만 그렇지않음.

해결: 원-핫 인코딩 (0 또는 1로 구분, 이진특성)

* OneHot Encoder 사이킷런

• 범주형(text) → 숫자형 → 원핫 벡터 by Categorical Encoder

[희소행렬
밀집행렬 encoding = "onehot -dense"

2.5.4. 특성 스케일링

모든특성의 범위를 같도록 만들어 주는 방법

① min-max 스케일링 : 정규화

0 ~ 1 범위에 들도록 이동하고 스케일 조절

* 사이킷런 MinMaxScaler

feature_range 매개변수로 숫자변동가능

② 표준화

- 평균을 빼고 표준편차로 나누어 분포의 분산 1로 만들기, 상한.하한 X → 문제 0, 이상치 영향↓

* 사이킷런 StandardScaler 변환기.

2.5.5 변환 파이프라인

- 연속된 변환을 순서대로 처리 도와주는 pipeline 클래스

① `pipeline([이름, 추정기], (" ,), ...)` 입력받기.

② `pipeline.fit` 에서도 호출하여 `fit-transform()` 에서도 순서대로 호출

③ `fit()` 에서도 호출

* `DataFrameSelector` 을 이용해 필요한 특성들만 파이프라인에 입력가능

* `Feature Union` : 두가지의 pipeline 을 합쳐 호출

2.6 모델선택과 훈련

2.6.1 훈련세트에서 훈련해 평가하기.

- 훈련세트에 대해 오차를 계산했을 때 과소적합 / 과대적합 판단가능

→ 계속 새로운 모델을 찾아가면서 확인

2.6.2 교차검증을 사용한 평가.

· K-접교차 검증 : 매개변수에 효용함수를 기대 → MSE의 반대로 계산

· 앙상블 학습 모델 : 여러 모델들을 모아서 하나로 만들기.

∴ 이를 이용해 2-5개 가능성 있는 모델을 선정하는 것이 목적.

2.7 모델세부튜닝

2.7.1 그리드 탐색

* 사이킷런 `GridSearchCV`

: 탐색해본 후 하루 하이퍼파라미터와 시도해볼 값을 지정 후 평가. → 최적의 추정기를 얻을 수 있음.

연속된 10의 거듭제곱수 시도 가능

2.7.2 랜덤 탐색

· 비교적 적은 수의 조합 → 그리드 탐색

· 탐색공간 ↑ → 랜덤탐색. (임의의수를 지정해 그 만큼 평가 → 컴퓨터 자원제어)

2.7.3 앙장블 방법

: 최상의 모델들을 연결.

2.7.4 최상의 모델과 오차분석

: 문제에 대한 좋은 통찰을 얻음 ex. 각 특성들의 중요도.

→ 오차에 대해 해결. (추가특성, 특성제거, 이상치 제거...)

2.7.5 테스트 세트로 시스템 평가.

테스트세트 → full-pipeline → 데이터 변환 → 최종모델평가. (성능이 더 낮은 것이 보통)

하이퍼 파라미터 변경하지 않기!

2.8 런칭, 모니터링, 그리고 시스템 유지보수.

① 입력데이터 연결

② 실시간 성능 체크 & 알람통지 하는 모니터링 코드

③ 시스템의 예측을 샘플링하여 평가.

④ 입력데이터 품질 평가.

⑤ 정기적으로 모델 훈련 필요.