핸즈온 머신러닝

PART1. 머신러닝

- 머신러닝이란? 데이터에서부터 학습하도록 컴퓨터를 프로그래밍하는 과학

ex) 스팸필터: 전통적인 프로그래밍 방식의 스팸 필터는 스팸 메일 발송자가 스팸 필터에 대항해 계속 단어를 바꾸면 엉원히 새로운 규칙을 추가해야 한다. 그러나 머신러닝 기반의 스팸필터는 스팸으로 지정한 메일에 특정 단어가 자주 나타는 것을 자동으로 인식하고 별도의 작업을 하지 않아도 자동으로 이 단어를 스팸으로 분류한다.

- 데이터마이닝: 머시러닝을 통해 배울 수도 있다. 머신러닝 기술을 적용해서 대용량의 데이터를 분석하면 겉으로 보이지 않는 패턴을 발견할 수 있다.

핸즈온 머신러닝

PART1. 머신러닝

- **애플리케이션 사례**

1. 생산 라인에서 제품 이미지를 분석해 자동으로 분류하기: 이미지 분류 작업.
   일반적으로 합성곱 신경망(CNN)을 사용하여 수행한다.

2. 자동으로 뉴스 기사 분류하기: 자연어 처리(NLP) 작업. 구체적으로 말하면 텍스트 분류이다.
   순환 신경망(RNN),CNN,트랜스포머를 사용해 해결할 수 있다.

3. 챗봇 또는 개인 비서 만들기: 자연어 이해(Natural Language Understanding: NLU)와 질문-대답 모듈을 포함해 여러가지 NLP 컴포넌트가 필요하다.

4. 다양한 성능지표를 기반으로 회사의 내년도 수익을 예측하기: 회귀 작업(즉 숫자로 값을 예측). 선형회귀나 다항회귀, 회귀 SVM, 회귀 랜덤 포레스트, 인공 신경망 등과 같은 회귀 모델을 사용하여 해결. 지난 성능 지표의 시퀀스를 고려한다면 RNN,CNN, 또는 트랜스포머 사용 가능

5. 음성 멍령에 반응하는 앱 만들기: 음싱 인식 작업. 오디오 샘플 처리. 길고 복잡한 시퀀스이므로
일반적으로 RNN,CNN 또는 트랜스포머를 사용한다.

핸즈온 머신러닝

PART1. 머신러닝

- **애플리케이션 사례**

6. 신용 카드 부정 거래 감지: 이상치 탐지 작업

7. 구매 이력을 기반으로 고객을 나누고, 각 집합마다 다른 마케팅 전략 계획: 군집 작업

8. 고차원의 복잡한 데이터셋을 명확하고 의미 있는 그래프로 표현하기: 데이터 시각화 작업. 차원축소 기법을 많이 사용한다.

9. 과거 구매 이력을 기반으로 고객이 관심을 가질 수 있는 상품 추천하기: 추천 시스템으로 과거 구매 이력을 (그리고 고객에 관한 다른 정보를) 인공 신경망에 주입하고 다음에 구매할 가능성이 가장 높은 상품을 출력하는 것이 한 가지 방법이다.

10. 지능형 게임 봇 만들기: 보통 강화학습으로 해결. 시간이 지나면 주어진 환경에서 보상이 최대가 되는 행동을 선택하는 (봇과 같은) 에이전트를 훈련하는 머신러닝의 한 분야이다.

핸즈온 머신러닝

PART1. 머신러닝

- **머신러닝 시스템의 종류**

1. 사람의 감독 하에 훈련하는 것인지 그렇지 않은 것인지 (지도/ 비지도/ 강화 학습)
2. 실시간으로 점진적인 학습을 하는지 아닌지 (온라인 학습과 배치 학습)
3. 단순하게 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 아니면 과학자들이 하는 것처럼 훈련 데이터 셋에서 패턴을 발견하여 예측 모델을 만드는 것인지(사례 기반 학습과 모델 기반 학습)

- **학습하는 동안의 감독 형태나 정보량**

지도학습, 비지도학습, 준지도학습, 강화학습

1. 지도학습: 알고리즘에 주입하는 훈련 데이터에 레이블이라는 원하는 답이 포함된다.
- 분류: 전형적인 지도 학습 작업.
- 회귀: 예측 변수라 부르는 특성을 사용하여 중고차 가격 같은 타깃 수치를 예측하는 것.

+ 속성: 데이터 타입
+ 특성: 속성과 값이 합쳐진 것을 의미

⇒ 일부 회귀 알고리즘은 분류에 사용할 수 있고, 반대도 가능하다. 대표적인 예로는 로지스틱 회귀가 있다.

핸즈온 머신러닝

PART1. 머신러닝

- **학습하는 동안의 감독 형태나 정보량**

지도학습, 비지도학습, 준지도학습, 강화학습

지도학습의 대표적인 학습 알고리즘
1) K- 최근접 이웃 2) 선형 회귀 3) 로지스틱 회귀 4) 서포트 벡터 머신 5) 결정 트리와 랜덤 포레스트 6) 신경망

2. 비지도학습: 훈련 데이터에 레이블이 없다. 시스템이 아무런 도움 없이 학습해야 한다.

비지도학습의 대표적인 학습 알고리즘
- 군집: 1) K-평균 2) DBSCAN 3) 계층 군집 분석 4) 이상치 탐지와 특이치 탐지 4) 원-클래스 5) 아이솔레이션 포레스트
- 시각화와 차원축소: 너무 많은 정보를 잃지 않으면서 데이터를 간소화한다. 상관관계가 있는 여러 특성을 하나로 합친다. 이를 특성 추출이라고 한다. 지도 학습 알고리즘 같은 머신러닝 알고리즘에 데이터를 주입하기 전에 차원 축소 알고리즘을 사용하여 훈련 데이터의 차원을 줄이는 것이 유용할 때가 많다. 실행 속도가 빨라지고 디스트와 메모리를 차지하는 공간도 줄고 경우에 따라 성능이 좋아지기도 한다.
  1) 주성분 분석(PCA) 2) 커널 PCA 3) 지역적 선형 인베딩 5) t-SNE
- 연관 규칙 학습: 대량의 데이터에서 특성 간의 흥미로운 관계를 찾는다.
  1) 어프라이어리 2) 이클렛

핸즈온 머신러닝

PART1. 머신러닝

- **학습하는 동안의 감독 형태나 정보량**

지도학습, 비지도학습, 준지도학습, 강화학습

3. 준지도 학습: 데이터에 레이블을 다는 것은 일반적으로 시간과 비용이 많이 들기 때문에 레이블이 없는 샘플이 많고 레이블된 샘플은 적은 경우가 많습니다. 어떤 알고리즘은 일부만 레이블이 있는 데이터를 다룰 수 있다.
대부분의 준지도 학습 알고리즘은 지도 학습과 비지도 학습의 조합으로 이루어져 있다. 예를 들어 심층 신뢰 신경망(DBN)은 여러 겹으로 쌓은 제한된 볼츠만 머신(RBM)이라 불리는 비지도 학습에 기초한다. RBM이 비지도 학습 방식으로 순차적으로 훈련된 다음 전체 시스템이 지도 학습 방식으로 세밀하게 조정된다.

4. 강화학습: 학습하는 시스템을 에이전트라고 부르며 환경을 관찰해서 행동을 실행하고 그 결과로 보상 혹은 부정적인 보상에 해당하는 벌점을 받는다. 시간이 지나면서 가장 큰 보상을 얻기 위해 정책이라고 부르는 최상의 전략을 스스로 학습한다.

핸즈온 머신러닝

PART1. 머신러닝

- **머신러닝 시스템의 종류**

1. 사람의 감독 하에 훈련하는 것인지 그렇지 않은 것인지 (지도/ 비지도/ 강화 학습)
2. 실시간으로 점진적인 학습을 하는지 아닌지 (온라인 학습과 배치 학습)
3. 단순하게 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 아니면 과학자들이 하는 것처럼 훈련 데이터 셋에서 패턴을 발견하여 예측 모델을 만드는 것인지(사례 기반 학습과 모델 기반 학습)

- **입력 데이터의 스트림부터 점진적으로 학습할 수 있는지**

배치학습과 온라인 학습

1. 배치 학습: 시스템이 점진적으로 학습할 수 없고, 가용한 데이터를 모두 사용해 훈련시켜야 한다.

시간과 자원을 많이 소모하므로 보통 오프라인에서 수행된다. 먼저 시스템을 훈련시키고 그런 다음 제품 시스템에 적용하면 더 이상의 학습없이 실행된다. 즉 학습한 것을 단지 적용만 한다=> 오프라인 학습

데이터를 업데이트하고 시스템의 새 비전을 필요한 만큼 자주 훈련시키면 되기 때문에 머신러닝 시스템에서도 변화에 적응할 수 있다.

그러나 데이터 양이 아주 많으면 매일 처음부터 새로 훈련시키도록 시스템을 자동화하는 것이 큰 비용이 발생하여 배치 학습 알고리즘을 사용하는게 불가능할 수도 있다.

핸즈온 머신러닝

PART1. 머신러닝

- **입력 데이터의 스트림부터 점진적으로 학습할 수 있는지**

배치학습과 온라인 학습

2. 온라인 학습: 데이터를 순차적으로 한 개씩 또는 미니배치라 부르는 작은 묶음 단위로 주입하여 시스템을 훈련시킨다. 매 학습 단계가 빠르고 비용이 적게 들어 시스템은 데이터가 도착하는 대로 즉시 학습할 수 있다.
연속적으로 데이터를 받고 빠른 변화에 스스로 적응해야하는 시스템에 적합하다. 온라인 학습 시스템이 새로운 데이터 샘플을 학습하면 학습이 끝난 데이터는 더는 필요하지 않으므로 버리면 된다. 그러면 많은 공간을 절약할 수 있다.

+ 학습률: 중요한 파라미터 중 하나로 변화하는 데이터에 얼마나 빠르게 적응할 것 인지. 학습률을 높게 하면 시스템이 데이터에 빠르게 적응하지만 예전 데이터를 금방 잊어버리고, 반대로 낮으면 시스템의 관성이 더 커져서 더 느리게 학습하게 된다. 그러나 새로운 데이터에 있는 작음이나 대표성 없는 데이터 포인트에 덜 민감해진다.

+온라인 학습에서 가장 큰 문제점: 시스템에 나쁜 데이터가 주입되었을 때 시스템 성능이 점진적으로 감소한다. 이런 위험을 줄이기 위해 시스템을 면밀히 모니터링하고 성능 감소가 감지되면 즉각 학습을 중지시켜야 한다. 입력 데이터를 모니터링해서 비정상 데이터를 잡아낼 수도 있다.(이상치 탐지 알고리즘 사용)

핸즈온 머신러닝

PART1. 머신러닝

- **머신러닝 시스템의 종류**

1. 사람의 감독 하에 훈련하는 것인지 그렇지 않은 것인지 (지도/ 비지도/ 강화 학습)
2. 실시간으로 점진적인 학습을 하는지 아닌지 (온라인 학습과 배치 학습)
3. 단순하게 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 아니면 과학자들이 하는 것처럼 훈련 데이터 셋에서 패턴을 발견하여 예측 모델을 만드는 것인지(사례 기반 학습과 모델 기반 학습)

- **머신러닝 시스템은 어떻게 일반화되는가**

사례 기반 학습과 모델 기반 학습

1. 사례 기반 학습: 시스템이 훈련 샘플을 기억함을써 학습한다. 그리고 유사도 측정을 사용해 새로운 데이터와 학습한 샘플을 (또는 학습한 샘플 중 일부를) 비교하는 식으로 일반화한다.
2. 모델 기반 학습: 샘플들의 모델을 만들어 예측에 사용한다.

1) 모델 선택(ex. 1인당 GDP의 선형 함수로 삶의 만족도 모델링)
여기서 선형 함수의 절편과 기울기가 두 개의 모델 파라미터에 해당하고,
이 모델 파라미터를 조정하여 어떤 선형함수를 표현하는 모델을 얻을 수 있다.

모델을 사용하기 전에 두 파라미터를 정의하여, 측정 지표를 통해 최상의 성능을 내도록 하는 값을 찾아야 한다.

핸즈온 머신러닝

PART1. 머신러닝

- **머신러닝 시스템은 어떻게 일반화되는가**

사례 기반 학습과 모델 기반 학습

- 측정 지표: 효용 함수(모델이 얼마나 좋은지 측정) 와 비용함수(모델이 얼마나 나쁜지 측정)로 정의할 수 있다.

선형회귀에서는 보통 선형 모델의 예측과 훈련 데이터 사이의 거리를 재는 비용함수를 사용한다.
이 거리를 최소화하는 것이 목표이다. 여기에서 선형 회귀 알고리즘을 훈련 데이터에 공급하여 데이터에 가장 잘 맞는 선형 모델의 파라미터를 찾는다. 이를 모델을 훈련시킨다고 말한다.

[전형적인 머신러닝 프로젝트의 형태]

1. 데이터 분석
2. 모델 선택
3. 훈련 데이터로 모델 훈련(학습 알고리즘이 비용함수를 최소화하는 모델 파라미터 찾기)
4. 새로운 데이터에 모델을 적용해 예측을 하고, 이 모델이 일반화되길 기대

핸즈온 머신러닝

PART1. 머신러닝

- 머신러닝의 주요 도전 과제(나쁜 데이터, 나쁜 알고리즘)

1. 충분하지 않은 양의 훈련 데이터
2. 대표성이 없는 훈련 데이터: 일반화가 잘되려면 우리가 일반화하기 원하는 새로운 사례를 훈련 데이터가 잘 대표하는 것이 중요하다.

Ex) 특히 매우 가난하거나 부유한 나라를 훈련 데이터로 사용할 경우 대표성이 없어 정확한 예측을 하지 못한다.

- 샘플이 작으면 샘플링 작음(우연에 의한 대표성 없는 데이터)
- 샘플이 커도 샘플링 편향(표본 추출 방법이 잘못되어 대표성을 띠지 못하는 데이터)

3. 낮은 품질의 데이터: 훈련 데이터가 에러, 이상치, 잡음으로 가득하다면 머신러닝 시스템이 내재된 패턴을 찾기 어려워 잘 작동하지 않을 것이다. 훈련 데이터 정제에 시간을 투자를 많이 해야한다.
4. 관련 없는 특성: 훈련 데이터에 관련 없는 특성이 적고 관련 있는 특성이 충분해야 시스템이 학습할 수 있다. 훈련에 사용할 좋은 특성들을 찾는 것이 성공적인 머신러닝 프로젝트의 핵심 요소이고 이를 특성 공학이라고 한다.

- 특성 선택/ 특성 추출(특성을 결합하여 더 유용한 특성을 만든다)

핸즈온 머신러닝

PART1. 머신러닝

- 머신러닝의 주요 도전 과제(나쁜 데이터, 나쁜 알고리즘)

5. 훈련 데이터 과대적합: 훈련 데이터에 너무 잘 맞지만 일반성이 떨어진다는 뜻
훈련 데이터에 있는 잡음의 양에 비해 모델이 너무 복잡할 때 일어난다.

해결방법)
1) 파라미터 수가 적은 모델을 선택(고차원 다항 모델-> 선형 모델)
2) 훈련 데이터에 있는 특성 수를 줄이거나, 모델에 제약을 가하여 단순화(규제)
3) 훈련 데이터를 더 많이 모은다.
4) 훈련 데이터의 잡음을 줄인다(예를 들어 오류 데이터 수정과 이상치 제거)

선형 모델의 절편과 기울기라는 두 개의 자유도를 학습 알고리즘에 부여한다. 우리가 기울기에 0이 되도록 강제하면 알고리즘에 한 개의 자유도만 남게 되고 데이터에 적절하게 맞춰지기 힘들 것이다. 즉 할 수 있는 것이 훈련 데이터에 가능한 한 가깝게 되도록 직선을 올리거나 내리는 것이 전부이므로 결국 평균 근처가 된다.
알고리즘이 기울기를 수정하도록 허락하되 작은 값을 갖도록 유지시키면 학습 알고리즘이 자유도1과 2사이의 적절한 어디나에 위치할 것이다. 이는 자유도 2인 모델보다는 단순하고 자유도 1인 모델보다는 복잡한 모델을 만든다.

=> 데이터에 완벽히 맞추는 것과 일반화를 위해 단순한 모델을 유지하는 것 사이의 올바른 균형을 찾는 것이 필요

핸즈온 머신러닝

PART1. 머신러닝

- 머신러닝의 주요 도전 과제(나쁜 데이터, 나쁜 알고리즘)

하이퍼파라미터: 학습 알고리즘의 파라미터. 학습하는 동안 적용할 규제의 양은 하이퍼파라미터가 결정한다. 학습 알고리즘으로부터 영향을 받지 않으며, 훈련 전에 미리 지정되고 훈련하는 동안에는 상수로 남아 있다.
규제 하이퍼 파라미터를 매우 큰 값으로 지정하면 (기울기가 0에 가까운) 거의 평편한 모델을 얻게 된다.

과소적합: 이는 모델이 너무 단순해서 데이터의 내재된 구조를 학습하지 못할 때 일어난다.
해결방법)
1) 모델 파라미터가 더 많은 강력한 모델 선택
2) 학습 알고리즘에 더 좋은 특성을 제공(특성 공학)
3) 모델의 제약 줄이기(규제 하이퍼파라미터 감소)

핸즈온 머신러닝

PART1. 머신러닝

- 테스트와 검증

1) 훈련 데이터를 훈련 세트와 테스트 세트 두 개로 나누어 모델을 훈련하고, 테스트한다.
새로운 샘플에 대한 오류 비율을 일반화 오차(외부 샘플 오차)라고 하며, 테스트 세트에서 모델을 평가함으로써 이 오차에 대한 추정값을 얻는다. 이 값은 이전에 본 적 없는 새로운 샘플에 모델이 얼마나 잘 작동하는지 알려준다. 훈련 오차가 낮지만 일반화 오차가 높다면 이는 모델이 훈련 데이터에 과대적합되었다는 것이다.

2) 그러나 일반화 오차를 테스트 세트에서 여러 번 측정하면 모델과 하이퍼파라미터가 테스트 세트에 최적화된 모델을 만든다. 이 문제에 대한 일반적인 해결 방법은 홀드아웃 검증이다. 간단하게 훈련 세트의 일부를 떼어내어 여러 후보 모델을 평가하고 가장 좋은 하나를 선택한다.

3) 전체 훈련 세트에서 검증 세트를 뺀 데이터에서 다양한 하이퍼파라미터 값을 가진 여러 모델을 훈련하여, 검증 세트에서 가장 높은 성능을 내는 모델을 선택한다.

4) 홀드아웃 검증 과정이 끝나면 이 최선의 모델을(검증 세트를 포함한) 전체 훈련 세트에서 다시 훈련하여 최종 모델을 만든다.

=> 작은 검정 세트를 여러 개를 사용해 반복적인 교차 검증을 수행. 검증 세트마다 나머지 데이터에서 훈련한 모델을 해당 검증세트에서 평강한다.