

생성형AI

Day 12

머신러닝 IV



목차

1. 양상을 기법
2. 추천 알고리즘
3. 강화 학습
4. 하이퍼 파라미터 튜닝
 1. 그리드 서치
 2. 랜덤 서치
5. 머신러닝과 윤리
6. 실습과제



앙상블 기법

앙상블 기법?

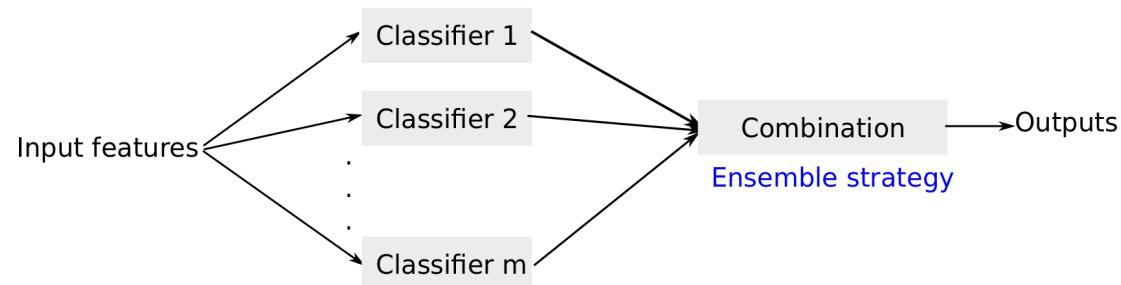
- 여러 개의 예측 모델을 결합하여 단일 모델보다 더 나은 성능을 얻는 방법

앙상블 기법의 목적?

- 예측 성능 향상: 여러 모델을 결합하여 예측 정확도 향상
- 과적합 감소: 다양한 모델의 결과를 결합하여 과적합 방지
- 안정성 향상: 모델의 변동성을 줄이고, 예측의 일관성 향상

앙상블 기법의 종류?

- 배깅(Bagging)
- 부스팅(Boosting)
- 스태킹(Stacking)



<https://ohdsi.github.io/PatientLevelPrediction/articles/BuildingEnsembleModels.html>

앙상블 기법

배깅(Bagging)

- 배깅(Bootstrap Aggregating)은 여러 모델을 병렬적으로 학습하고, 예측을 평균, 다수결 투표를 통해 결정하는 방법

배깅의 원리

1. 부트스트랩 샘플링

- 원본 데이터셋에서 중복을 허용하여 여러 개의 샘플을 무작위로 추출

2. 개별 모델 학습

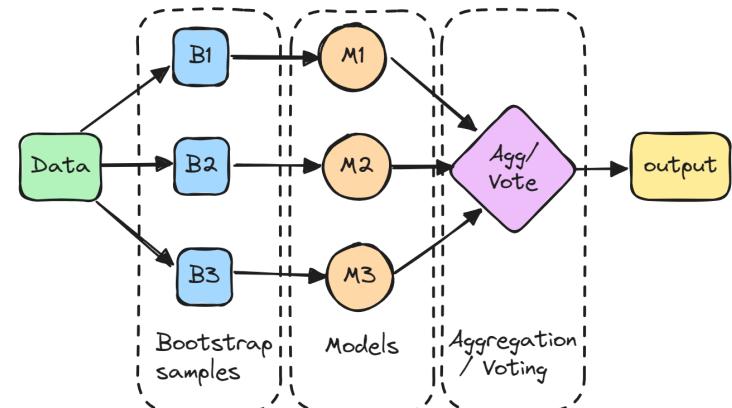
- 각 부트스트랩 샘플을 사용하여 개별 모델을 학습

3. 예측 결합

- 모든 모델의 예측을 평균내거나, 다수결 투표를 통해 최종 예측 결정

대표 알고리즘

- 랜덤 포레스트



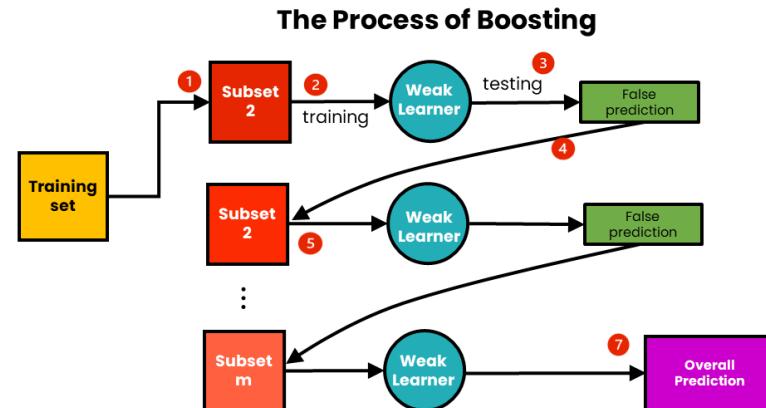
앙상블 기법

부스팅(Boosting)

- 부스팅은 순차적으로 모델을 학습시키며, 이전 모델이 잘못 예측한 샘플에 더 큰 가중치를 부여하여 오류를 보정하는 방법

부스팅의 원리

1. 초기 모델 학습
 - 첫 번째 모델을 학습
2. 오류 샘플 가중치 증가
 - 첫 번째 모델이 잘못 예측한 샘플의 가중치를 증가
3. 순차적 모델 학습
 - 가중치가 조정된 샘플을 사용하여 다음 모델을 학습
4. 예측 결합
 - 모든 모델의 예측을 가중 평균하여 최종 예측



대표 알고리즘

- AdaBoost, Gradient Boosting, XGBoost

앙상블 기법

스태킹(Stacking)

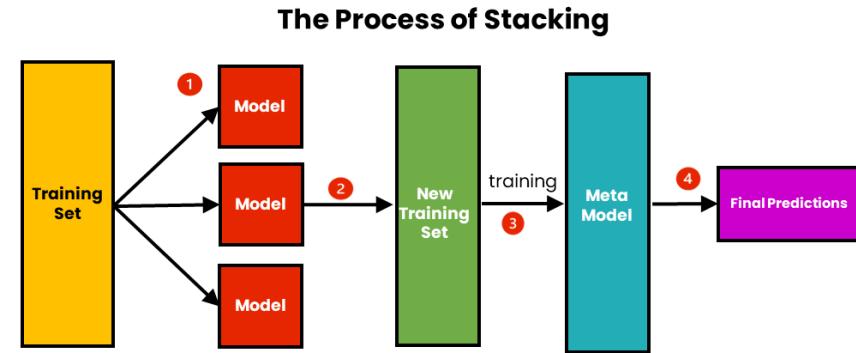
- 스태킹은 여러 모델의 예측 결과를 입력으로 사용하여 메타 모델을 학습시키는 방법

스태킹의 원리

1. 기본 모델 학습
 - 여러 개의 기본 모델을 학습
2. 메타 모델 학습
 - 기본 모델의 예측 결과를 입력으로 사용하여 메타 모델을 학습
3. 최종 예측
 - 메타 모델의 예측 결과를 최종 예측으로 사용

대표 알고리즘

- 다층 퍼셉트론(MLP)을 메타 모델로 사용하는 스태킹 방법



https://medium.com/@brijesh_soni/stacking-to-improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28

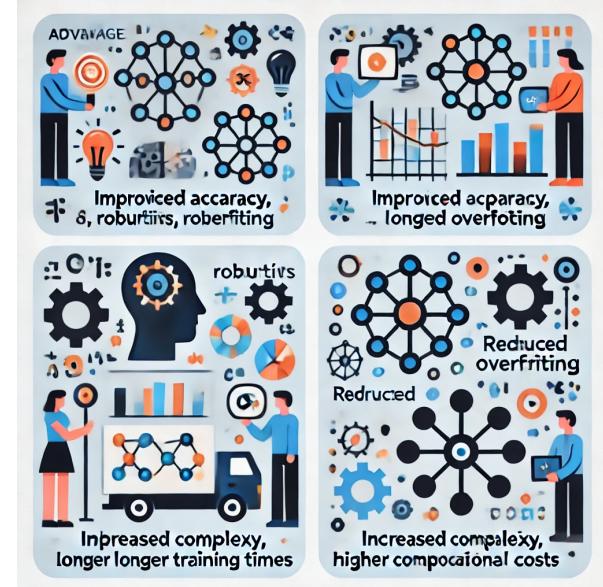
앙상블 기법

앙상블 기법 장점

- 예측 성능 향상: 여러 모델의 예측을 결합하여 더 높은 정확도를 얻음
- 과적합 감소: 다양한 모델의 결과를 결합하여 개별 모델의 과적합 방지
- 안정성 향상: 모델의 변동성을 줄이고, 예측의 일관성 향상

앙상블 기법 단점

- 복잡성 증가: 여러 모델을 학습시키고 결합하는 과정이 복잡할 수 있음
- 해석 어려움: 단일 모델에 비해 해석이 어려울 수 있음
- 계산 비용: 여러 모델을 학습시키야 하기 때문에 계산 비용이 증가



추천 시스템

추천 시스템?

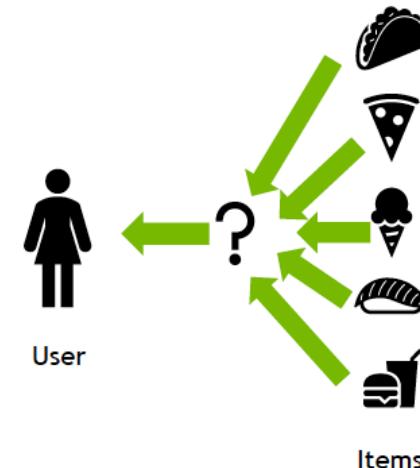
- 사용자와 아이템 간의 관계를 분석하여 사용자에게 적합한 아이템을 추천하는 시스템

추천 시스템의 목적?

- 사용자 만족도 향상: 사용자가 선호할 만한 아이템을 추천하여 만족도 향상
- 판매 증대: 적절한 제품 추천을 통해 구매를 촉진
- 사용자 참여 증대: 맞춤형 콘텐츠를 제공하여 사용자 참여를 유도

추천 시스템의 종류?

- 협업 필터링 (Collaborative Filtering)
- 콘텐츠 기반 필터링 (Content-Based Filtering)



추천 시스템

협업 필터링 (Collaborative Filtering)

- 협업 필터링은 사용자 간의 유사성 또는 아이템 간의 유사성을 이용하여 추천을 수행하는 방법

협업 필터링 원리

1. 사용자 기반 협업 필터링 (User-Based Collaborative Filtering)

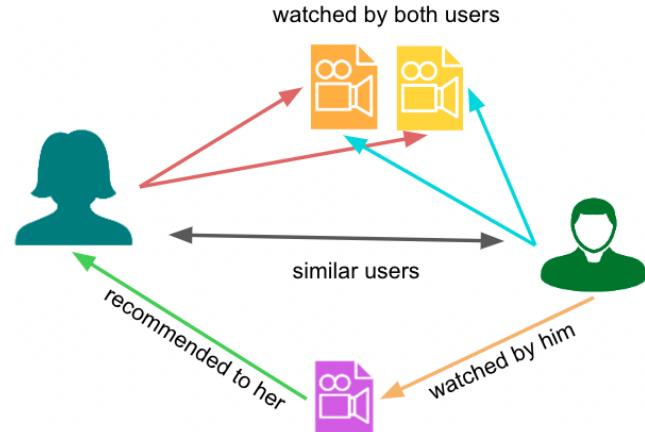
- 유사한 취향을 가진 사용자들이 선호하는 아이템을 추천
 - A와 B가 비슷한 영화를 좋아하면, A가 좋아하는 영화를 B에게 추천
- #### 2. 아이템 기반 협업 필터링 (Item-Based Collaborative Filtering)
- 유사한 아이템을 선호하는 사용자들에게 그 아이템을 추천
 - A가 좋아하는 영화와 비슷한 영화를 A에게 추천

협업 필터링 장점

- 개인화된 추천: 사용자의 과거 행동을 기반으로 추천하여 개인화된 경험 제공
- 데이터 활용: 대규모 사용자 데이터를 활용하여 추천 정확도 향상

협업 필터링 단점

- 콜드 스타트 문제: 신규 사용자나 신규 아이템에 대한 정보가 부족할 때 추천 어려움
- 데이터 희소성 문제: 사용자-아이템 매트릭스가 희소할 때 추천의 정확도 저하



추천 시스템

콘텐츠 기반 필터링 (Content-Based Filtering)

- 콘텐츠 기반 필터링은 아이템의 특징을 분석하여 사용자가 선호할 만한 유사한 아이템을 추천하는 방법

콘텐츠 기반 필터링의 원리

1. 아이템 특징 추출

- 각 아이템의 특징(장르, 배우, 저자 등)을 추출
- 예: 영화의 장르, 출연 배우, 감독 등

2. 사용자 프로필 생성

- 사용자가 선호하는 아이템의 특징을 기반으로 사용자 프로필 생성
- 예: 사용자가 좋아하는 영화의 장르와 출연 배우를 분석하여 사용자 프로필 생성

3. 유사한 아이템 추천

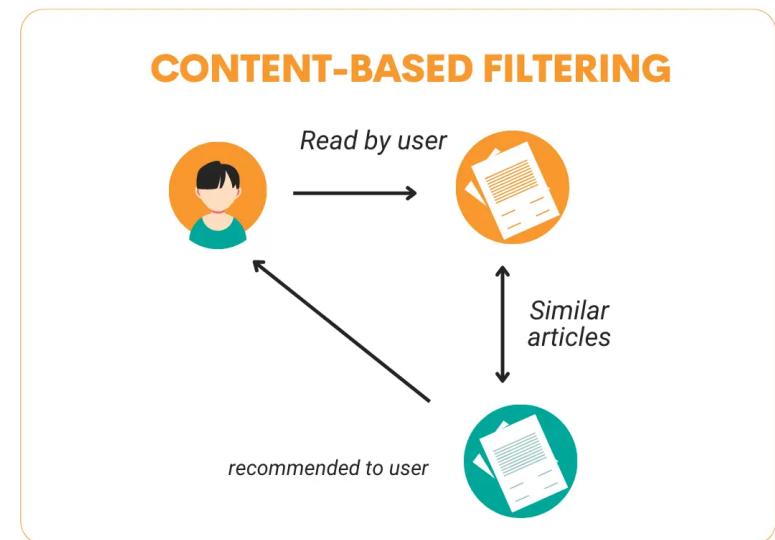
- 사용자 프로필과 유사한 특징을 가진 아이템을 추천
- 예: 사용자가 좋아하는 장르의 영화를 추천

콘텐츠 기반 필터링 장점

- 콜드 스타트 문제 해결: 사용자 또는 아이템의 특징만으로 추천 가능
- 투명성: 추천 이유를 설명하기 용이

콘텐츠 기반 필터링 단점

- 특징 선택의 어려움: 적절한 특징을 선택하는 것이 어려울 수 있음
- 다양성 부족: 사용자가 이미 알고 있는 유사한 아이템만 추천할 가능성



추천 시스템

하이브리드 방법 (Hybrid Methods)

- 하이브리드 방법은 협업 필터링과 콘텐츠 기반 필터링을 결합하여 추천 성능을 향상시키는 방법

하이브리드 방법의 원리

1. 결합 방법

- 협업 필터링과 콘텐츠 기반 필터링의 결과를 결합하여 최종 추천을 생성
- 예: 두 방법의 추천 결과를 가중 평균하여 결합

2. 단계적 방법

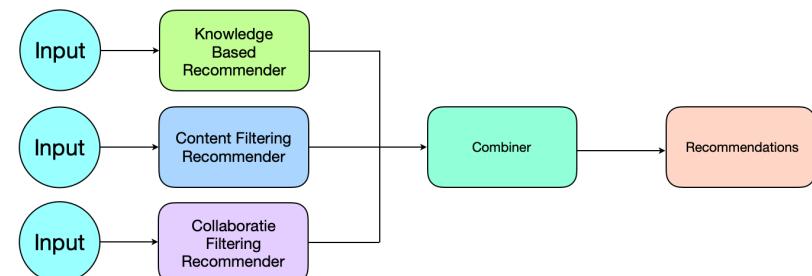
- 첫 번째 단계에서 협업 필터링을 사용하고, 두 번째 단계에서 콘텐츠 기반 필터링을 사용하여 추천
- 예: 협업 필터링으로 초기 후보를 선정하고, 콘텐츠 기반 필터링으로 최종 추천

하이브리드 방법 장점

- 성능 향상: 두 방법의 장점을 결합하여 추천 성능을 극대화
- 유연성: 다양한 방법을 조합하여 최적의 추천 시스템 구축 가능

하이브리드 방법 단점

- 복잡성 증가: 여러 방법을 결합하여 복잡성이 증가할 수 있음
- 구현 어려움: 다양한 방법을 효과적으로 결합하기 위한 구현이 어려울 수 있음



추천 시스템

추천 시스템의 주요 구성 요소

1. 사용자–아이템 매트릭스

- 사용자와 아이템 간의 상호작용(예: 평점, 클릭 등)을 기록한 매트릭스
- 예: 영화 평점 매트릭스

2. 특징 벡터

- 사용자와 아이템의 특징을 나타내는 벡터
- 예: 영화의 장르, 감독, 출연 배우 등

3. 유사도 계산

- 사용자 간 또는 아이템 간의 유사도를 계산하는 방법
- 예: 코사인 유사도(Cosine Similarity), 피어슨 상관계수(Pearson Correlation)

4. 모델 학습 및 평가

- 추천 모델을 학습시키고, 성능을 평가하는 방법
- 예: RMSE(Root Mean Squared Error), MAE(Mean Absolute Error), Precision, Recall
- 추천 시스템의 구성, 목적에 따라 다양한 평가 방법 존재

추천 시스템

추천 시스템의 평가 방법

1. 정확도 (Accuracy)

- 추천 시스템이 얼마나 정확하게 아이템을 추천하는지를 평가
- 예: Precision, Recall, F1 Score

2. 다양성 (Diversity)

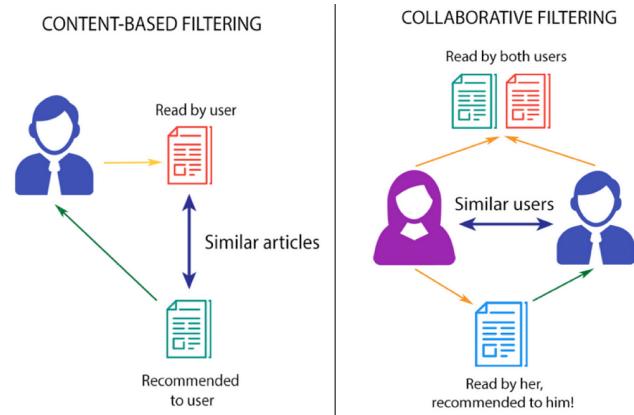
- 추천된 아이템이 얼마나 다양한지를 평가
- 예: 사용자에게 새로운 아이템을 추천할 수 있는 능력

3. 신뢰성 (Trustworthiness)

- 추천된 아이템이 사용자에게 얼마나 신뢰성을 가지는지를 평가
- 예: 추천된 아이템의 설명 가능성

4. 사용자 만족도 (User Satisfaction)

- 추천 시스템을 사용한 후 사용자의 만족도를 평가
- 예: 사용자 피드백, 설문 조사



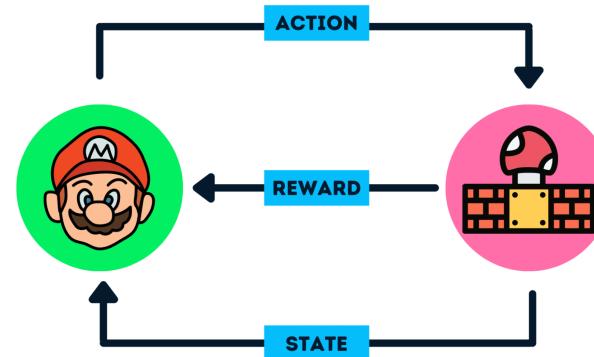
강화 학습

강화 학습?

- 강화학습은 에이전트가 환경과 상호작용하며, 보상을 최대화하는 행동 정책을 학습하는 방법

강화 학습의 목적?

- 주어진 환경에서 최적의 행동 정책을 학습하여 누적 보상을 최대화하는 것



<https://www.kdnuggets.com/2022/05/reinforcement-learning-newbies.html>

강화 학습

Q-learning

- 상태-행동 가치 함수(Q-함수)를 학습하여 최적의 정책을 찾는 방법

Q-learning 원리

1. 초기화

- 모든 상태-행동 쌍의 Q-값을 초기화 (보통 0으로 초기화)

2. 에이전트-환경 상호작용

- 에이전트는 현재 상태에서 행동을 선택하고, 환경으로부터 보상과 다음 상태를 관찰

3. Q-함수 업데이트

- Q-함수를 다음과 같이 업데이트

- α 는 학습률(Learning Rate), γ 할인 인자(Discount Factor)

4. 정책 업데이트

- Q-함수가 업데이트된 후, 에이전트는 새로운 정책에 따라 행동을 선택

5. 반복

- 위 과정을 에피소드 또는 시간 단계마다 반복하여 Q-함수를 수렴

$$\text{New } Q(s,a) = Q(s,a) + \alpha [R(s,a) + \gamma \max_{a'} Q'(s', a') - Q(s,a)]$$

Diagram illustrating the Q-learning update rule:

- New Q value for the state and action** (red box): The result of the update.
- Current Q values** (purple boxes): Inputs from the previous step.
- Reward for taking an action in a state** (orange box): Input from the environment.
- Maximum expected future reward** (green box): Input from the environment.
- Learning Rate** (α): Multiplies the difference between current and new Q-values.
- Discount Rate** (γ): Multiplies the maximum expected future reward.

강화 학습

Q-learning의 주요 구성 요소

학습률 (Learning Rate, α)

- Q-값 업데이트의 비율을 결정 ($0 < \alpha \leq 1$)
- 학습률이 높을수록 Q-값이 빠르게 업데이트

할인 인자 (Discount Factor, r)

- 미래 보상의 현재 가치를 결정 ($0 \leq r \leq 1$)
- 할인 인자가 클수록 미래 보상을 더 중요하게 고려

탐험과 활용 (Exploration vs. Exploitation)

- 탐험(Exploration): 새로운 행동을 시도하여 더 많은 정보를 얻는 과정
- 활용(Exploitation): 현재 알고 있는 최적의 행동을 선택하는 과정
- ϵ -탐욕정책 (ϵ -greedy policy): 확률 ϵ 로 탐험하고, $1 - \epsilon$ 의 확률로 최적의 행동을 선택

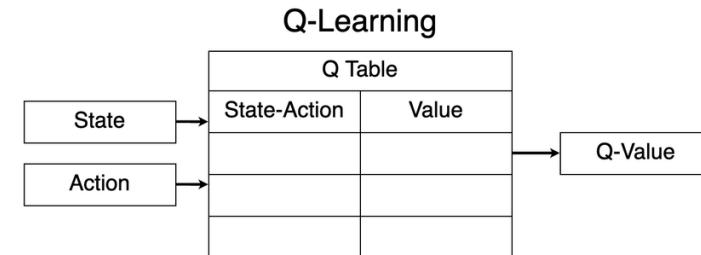
강화 학습

Q-learning의 장점

- 단순성: 알고리즘이 단순하고 구현이 용이
- 오프라인 학습: 환경의 모델이 필요 없으며, 오프라인으로 학습 가능
- 보편성: 다양한 강화학습 문제에 적용 가능

Q-learning의 단점

- 큰 상태 공간: 상태 공간이 크면 Q-테이블의 크기가 커져서 메모리와 계산 비용이 증가
- 연속적인 상태 및 행동 공간: Q-learning은 이산적인 상태 및 행동 공간에 적합하며, 연속적인 공간에서는 효율적이지 않음
- 탐험-활용 균형: 적절한 ϵ 값을 선택하는 것이 중요



하이퍼 파라미터 튜닝

하이퍼 파라미터 (Hyperparameter)?

- 모델 학습 전에 설정하는 값, 학습 과정 중 변경되지 않음

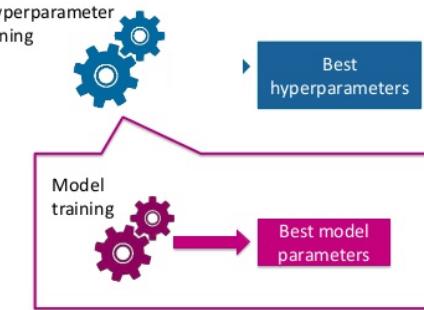
하이퍼 파라미터 예시

- 결정 트리의 최대 깊이
- SVM의 커널 종류
- 신경망의 학습률

하이퍼 파라미터 튜닝 (Hyperparameter Tuning)?

- 머신러닝 모델의 성능을 최적화하기 위해 하이퍼 파라미터의 최적값을 찾는 과정

Hyperparameter tuning vs. model training



하이퍼 파라미터 튜닝

하이퍼 파라미터 튜닝의 목적

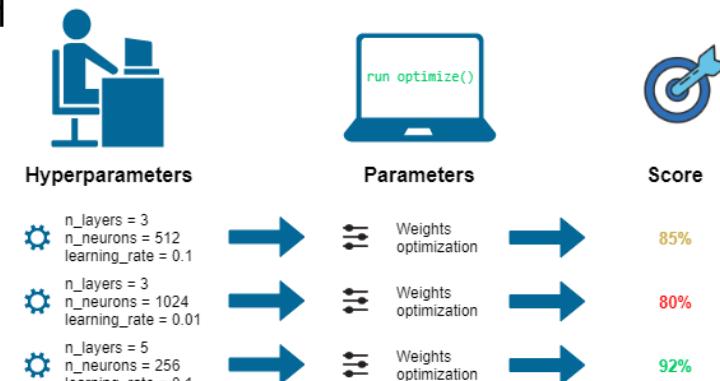
- 모델 성능 최적화: 하이퍼 파라미터의 최적값을 찾아 모델의 예측 성능을 극대화
- 과적합 방지: 적절한 하이퍼 파라미터 설정으로 과적합을 방지하고 일반화 성능을 향상
- 계산 비용 최적화: 효율적인 하이퍼 파라미터 설정으로 모델 학습에 소요되는 시간 절약

하이퍼 파라미터 튜닝과 모델 성능

- 하이퍼 파라미터 값에 따라 모델의 학습 속도와 성능이 크게 달라질 수 있음
- 잘못된 하이퍼 파라미터 설정은 과적합 또는 과소적합을 초래

하이퍼 파라미터 튜닝 방법

- 그리드 서치 (Grid Search)
- 랜덤 서치 (Random Search)



하이퍼 파라미터 튜닝

그리드 서치 (Grid Search)

- 하이퍼 파라미터의 가능한 값들을 모두 탐색하여 최적의 조합을 찾는 방법

그리드 서치 원리

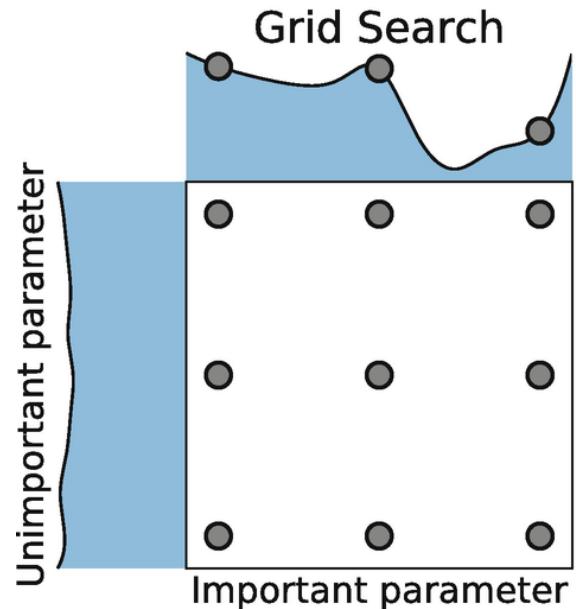
1. 하이퍼 파라미터 공간 정의
 - 각 하이퍼 파라미터에 대해 탐색할 값들의 범위를 정의
2. 조합 탐색
 - 가능한 모든 하이퍼 파라미터 조합에 대해 모델을 학습하고 평가
3. 최적 조합 선택
 - 평가 성능이 가장 좋은 하이퍼 파라미터 조합을 선택

그리드 서치 장점

- 모든 조합을 탐색하므로 최적의 하이퍼 파라미터를 찾을 가능성이 높음

그리드 서치 단점

- 계산 비용이 많이 들고, 시간 소모가 큼



Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." The Journal of Machine Learning Research 13.1 (2012): 281-305.

하이퍼 파라미터 튜닝

랜덤 서치 (Random Search)

- 하이퍼 파라미터 공간에서 무작위로 조합을 선택하여 탐색하는 방법

랜덤 서치 원리

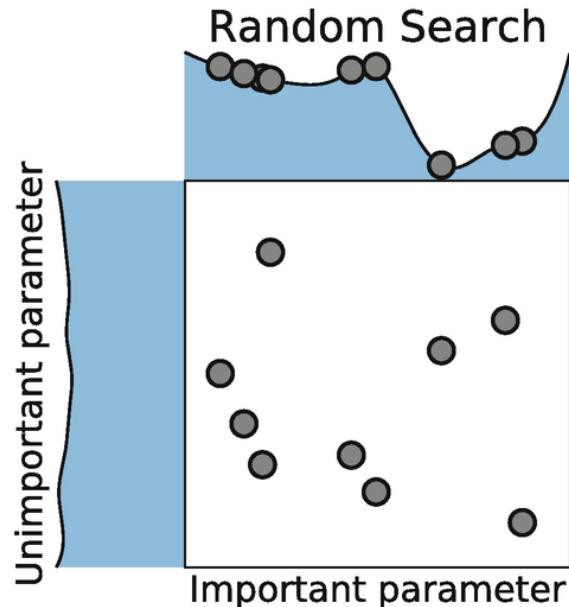
1. 하이퍼 파라미터 공간 정의
 - 각 하이퍼 파라미터에 대해 탐색할 값들의 범위를 정의
2. 무작위 조합 선택
 - 정의된 하이퍼 파라미터 공간에서 무작위로 조합을 선택
3. 최적 조합 선택
 - 무작위로 선택된 조합들 중에서 평가 성능이 가장 좋은 조합을 선택

그리드 서치 장점

- 계산 비용이 그리드 서치보다 적고, 빠르게 탐색 가능

그리드 서치 단점

- 무작위 선택이므로 최적의 하이퍼 파라미터를 찾지 못할 수도 있음



Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." The Journal of Machine Learning Research 13.1 (2012): 281-305.

머신러닝과 윤리

공정성

- 특정 그룹이나 개인에게 불공정한 결과를 초래하지 않도록 보장해야함
- 다양한 출처에서 데이터를 수집하여 데이터 다양성을 확보해야함

투명성

- 모델의 의사결정 과정을 이해하고 설명할 수 있어야함

프라이버시

- 개인의 데이터를 보호하고, 무단 사용을 방지해야함
- 데이터 비식별화가 필요함

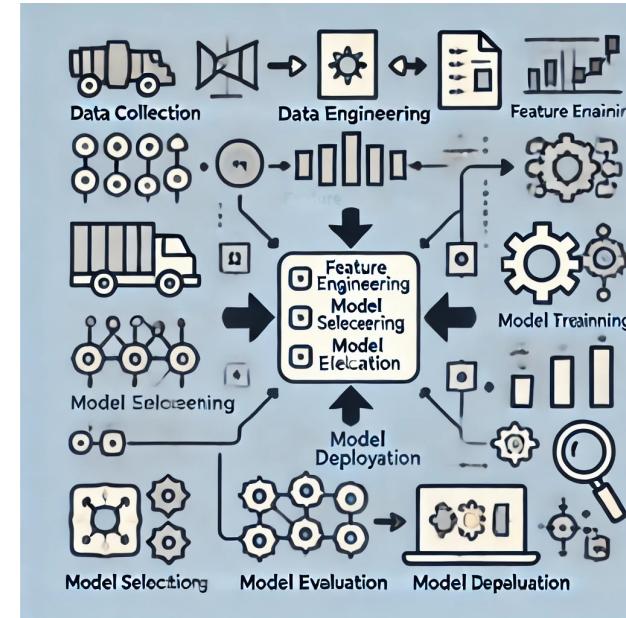
책임성

- 모델의 결과와 사용에 대한 책임을 져야함
- 모델의 성능과 윤리적 문제를 지속적으로 모니터링하고 개선할 수 있는 피드백 메커니즘을 마련해야함



머신러닝 시스템 설계

1. 요구사항 명확히 하기
2. 요구사항을 머신러닝 작업으로 구조화하기
3. 데이터 준비
4. 모델 개발
5. 평가 및 피드백
6. 배포 및 운영
7. 모니터링 및 서비스 피드백



머신러닝 시스템 설계

1. 요구사항 명확히 하기

- 우리 비즈니스의 목표가 무엇인가?

: 비즈니스의 수익상승? 서비스 이용자의 상승?

- 데이터 상태는 어떤가?

: 데이터소스? 데이터의 크기? 라벨 유무?

- 가용 가능한 리소스는 얼마나 되나?

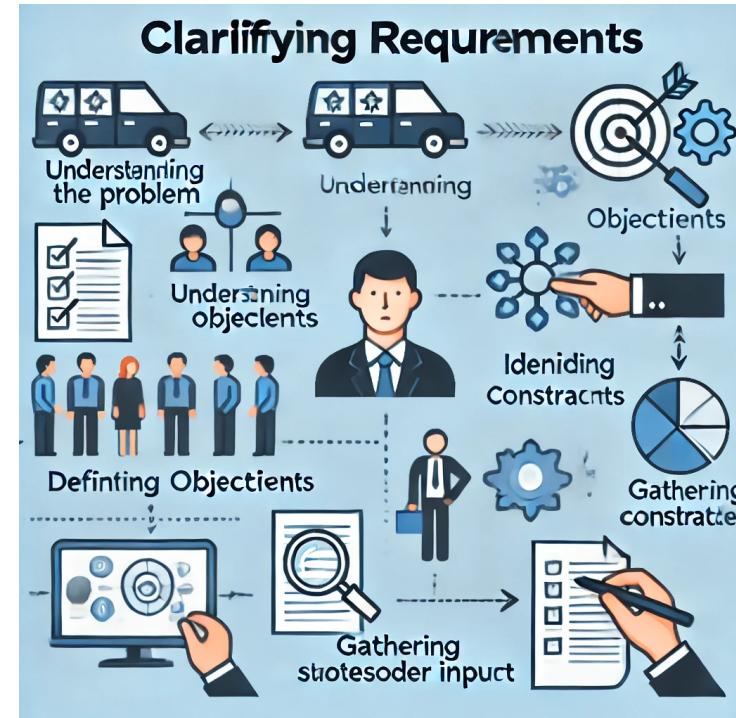
: 클라우드 환경? 모바일 환경? GPU 리소스 유무?

- 시스템의 규모는 얼마나 되나?

: 서비스 이용자 수? 입출력 컨텐츠 종류? 컨텐츠 크기?

- 기대하는 성능은?

: 목표 정확도? 실시간 서비스?



머신러닝 시스템 설계

2. 요구사항을 머신러닝 작업으로 구조화하기

- 머신러닝 목표 정의

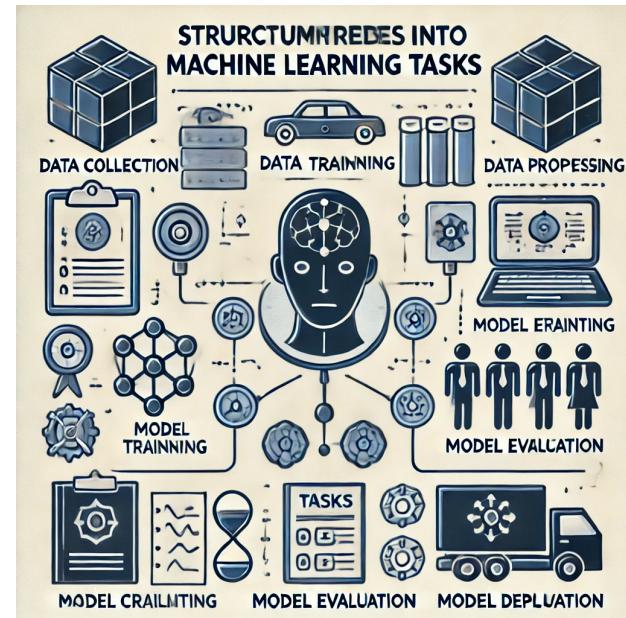
: 비즈니스 목표와 연관지어 머신러닝의 목표 정의, 비즈니스 목표를 이끌어 낼수 있는 것

- 시스템의 입출력 지정

: 이 목표를 수행하는 머신러닝 시스템의 입력, 출력값을 지정

- 적절한 머신러닝 선택

: 학습 유형, 모델 유형



머신러닝 시스템 설계

3. 데이터 준비

- 데이터 소스

: 누가 제공하는 데이터인가? 신뢰할 만한가?
어떻게 만들어졌나? 라벨링은 어떻게 처리?

- 데이터 저장소

: RDBMS, NoSQL

- 데이터 유형

: 텍스트? 정형데이터?

- 데이터 전처리

: 결측치 처리, 이상치 처리, 스케일링?



머신러닝 시스템 설계

4. 모델 개발

- 간단한 모델 부터 시작

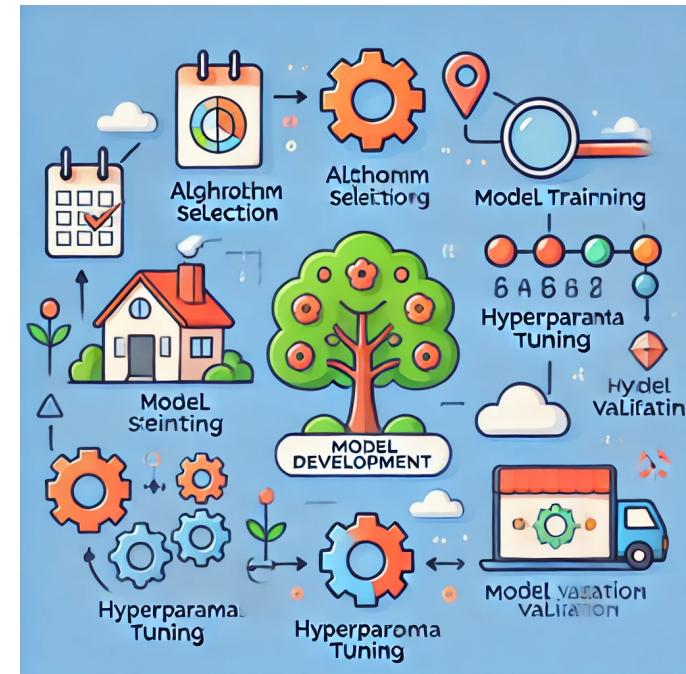
: 빠르게 개발, 테스트 가능한 모델

- 점차 복잡한 모델로 전환

: 복잡도가 높은 모델, 하이퍼 파라미터 조정

- 원한다면 여러 모델을 조합

: 앙상블



머신러닝 시스템 설계

5. 평가 및 피드백

- 오프라인 평가

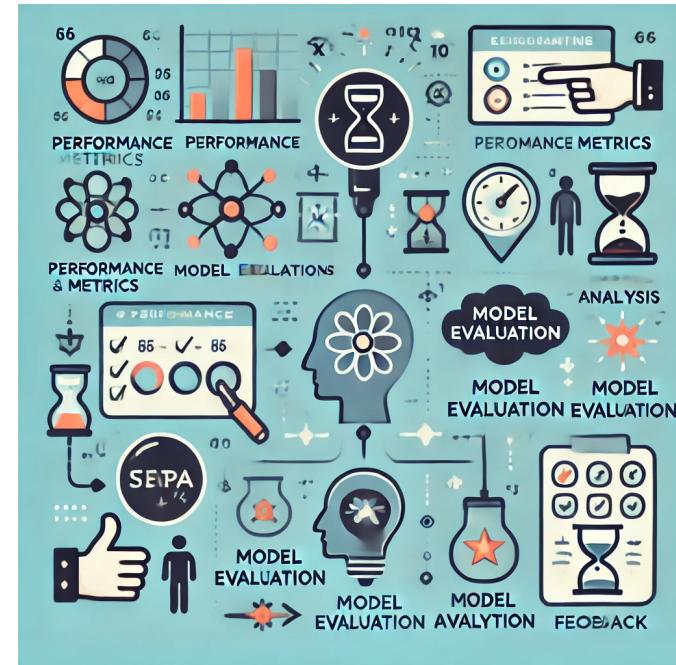
: confusion matrix, MSE

- 온라인 평가

: 클릭율, 체류시간 등

- 피드백

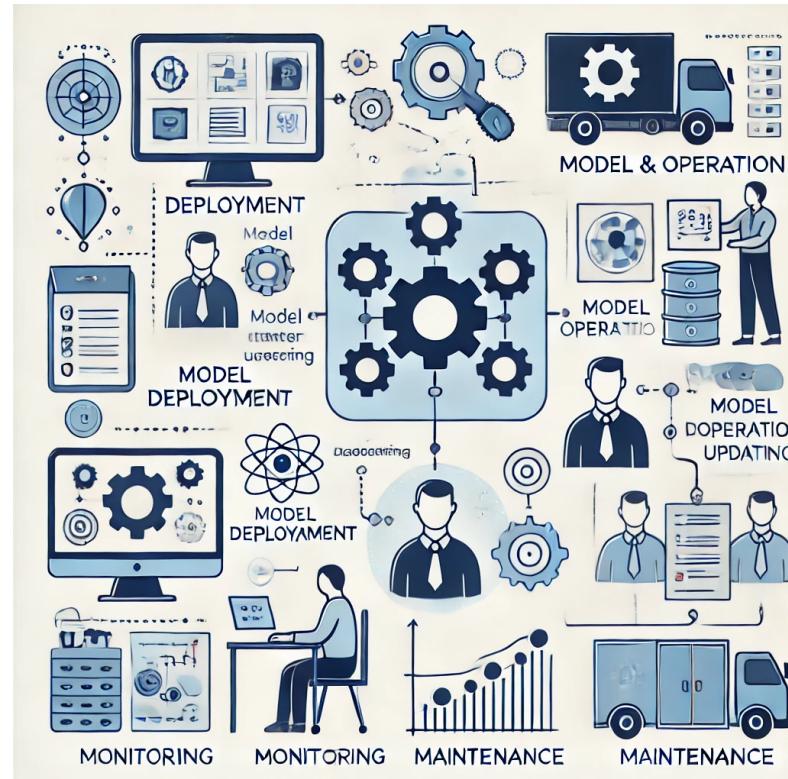
: 파라미터 조정 등



머신러닝 시스템 설계

6. 배포 및 운영

- 클라우드? 온디바이스?
: 배포위치, 서비스 제공 방법
- 모델 압축
: 자식증류, 가지치기, 양자화



머신러닝 시스템 설계

7. 모니터링 및 서비스 피드백

- 운영환경에서 테스트
: 세도배포, A/B테스트
- 예측값 파이프라인 구성
: 배치예측, 온라인 예측
- 모니터링 대상
: 운영서버(평균 서빙시간, 처리량, CPU/GPU사용률),
머신러닝관련(입출력 데이터 모니터링, 정확도)



이론 실습

colab: https://colab.research.google.com/drive/1vAGAGrONY5pckVUjsFKqBXh_KkP6RPlm?usp=sharing

실습 과제

데이터셋을 활용한 머신러닝 시스템 적용

1. 데이터셋 수집

2. 데이터 탐색 및 전처리

3. 모델 선택 및 학습

- 양상블
- 추천
- 하이퍼파라미터 튜닝

4. 모델 평가 및 시각화

실습 진행