# <머신러닝 프로젝트 체크리스트>

1. 문제를 정의하고 큰 그림을 그립니다.

2. 데이터를 수집합니다.

3. 통찰을 얻기 위해 데이터를 탐색합니다.

4. 데이터에 내재된 패턴이 머신러닝 알고리즘에 잘 드러나도록 데이터를 준비합니다.

5. 여러 다른 모델을 시험해보고 가장 좋은 몇 개를 고릅니다.

6. 모델을 세밀하게 튜닝하고 이들을 연결해 최선의 솔루션을 만듭니다.

7. 솔루션을 출시합니다.

8. 시스템을 론칭하고 모니터링, 유지 보수합니다. 당연히 이 체크리스트는 각자의 필요에 맞게 조정될 수 있습니다.

## B.1 문제를 정의하고 큰 그림을 그립니다.

**1. 목표를 비즈니스 용어로 정의합니다.**

**2. 이 솔루션은 어떻게 사용될 것인가?**

**3. (만약 있다면) 현재 솔루션이나 차선책은 무엇인가?**

**4. 어떤 문제라고 정의할 수 있나 (지도/비지도, 온라인/오프라인 등)?**

**5. 성능을 어떻게 측정해야 하나?**

**6. 성능 지표가 비즈니스 목표에 연결되어 있나?**

**7. 비즈니스 목표에 도달하기 위해 필요한 최소한의 성능은 얼마인가?**

**8. 비슷한 문제가 있나? 이전의 방식이나 도구를 재사용할 수 있나?**

**9. 해당 분야의 전문가가 있나?**

**10. 수동으로 문제를 해결하는 방법은 무엇인가?**

**11. 여러분이 (또는 다른 사람이) 세운 가정을 나열합니다.**

**12. 가능하면 가정을 검증합니다.**

## B.2 데이터를 수집합니다

(노트: 새로운 데이터를 쉽게 얻을 수 있도록 최대한 자동화하세요.)

**1. 필요한 데이터와 양을 나열합니다.**

**2. 데이터를 얻을 수 있는 곳을 찾아 기록합니다.**

**3. 얼마나 많은 공간이 필요한지 확인합니다.**

**4. 법률상의 의무가 있는지 확인하고 필요하다면 인가를 받습니다.**

**5. 접근 권한을 획득합니다.**

**6. 작업 환경을 만듭니다 (충분한 저장 공간으로).**

**7. 데이터를 수집합니다.**

**8. 데이터를 조작하기 편리한 형태로 변환합니다(데이터 자체는 바꾸지 않습니다).**

**9. 민감한 정보가 삭제되었거나 보호되었는지 검증합니다(예를 들면 개인정보 비식별화).**

**10. 데이터의 크기와 타입(시계열, 표본, 지리정보 등)을 확인합니다.**

**11. 테스트 세트를 샘플링하여 따로 떼어놓고 절대 들여다보지 않습니다(데이터 염탐 금지!).**

## B.3 데이터를 탐색합니다

(노트: 이 단계에서는 해당 분야의 전문가에게 조언을 구하세요.)

**1. 데이터 탐색을 위해 복사본을 생성합니다 (필요하면 샘플링하여 적절한 크기로 줄입니다).**

**2. 데이터 탐색 결과를 저장하기 위해 주피터 노트북을 만듭니다.**

**3. 각 특성의 특징을 조사합니다.**

- 이름
- 타입(범주형, 정수/부동소수, 최댓값/최솟값 유무, 텍스트, 구조적인 문자열 등)
- 누락된 값의 비율(%)
- 잡음 정도와 잡음의 종류(확률적, 이상치, 반올림 에러 등)
- 이 작업에 유용한 정도
- 분포 형태(가우시안, 균등, 로그 등)

**4. 지도 학습 작업이라면 타깃 속성을 구분합니다.**

**5. 데이터를 시각화합니다.**

**6. 특성 간의 상관관계를 조사합니다.**

**7. 수동으로 문제를 해결할 수 있는 방법을 찾아봅니다.**

**8. 적용이 가능한 변환을 찾습니다.**

**9. 추가로 유용한 데이터를 찾습니다(있다면 '데이터를 수집합니다'로 돌아갑니다)**

**10. 조사한 것을 기록합니다.**

## B.4 데이터를 준비합니다.

(노트:
- 데이터의 복사본으로 작업합니다(원본 데이터셋은 그대로 보관합니다)..
- 적용한 모든 데이터 변환은 함수로 만듭니다. 여기에는 다섯 가지 이유가 있습니다.
- 다음에 새로운 데이터를 얻을 때 데이터 준비를 쉽게 할 수 있기 때문입니다.
- 다음 프로젝트에 이 변환을 쉽게 적용할 수 있기 때문입니다.
- 테스트 세트를 정제하고 변환하기 위해서입니다.
- 솔루션이 서비스에 투입된 후 새로운 데이터 샘플을 정제하고 변환하기 위해서입니다.
- 하이퍼파라미터로 준비 단계를 쉽게 선택하기 위해서입니다.)

**1. 데이터 정제**

- 이상치를 수정하거나 삭제합니다 (선택사항).
- 누락된 값을 채우거나(예를 들면 00이나 평균, 중간값 등으로), 그 행(또는 열)을 제거합니다.

**2. 특성 선택(선택사항)**

- 작업에 유용하지 않은 정보를 가진 특성을 제거합니다.

**3. 적절한 특성 공학**

- 연속 특성 이산화하기
- 특성 분해하기(예를 들면 범주형, 날짜/시간 등)
- 가능한 특성 변환 추가하기(예를 들면 log(x), sqrt(x), x^2 등)
- 특성을 조합해 가능성 있는 새로운 특성 만들기

**4. 특성 스케일 조정표준화 또는 정규화**

## B.5 가능성 있는 몇 개의 모델을 고릅니다.

(노트: • 데이터가 매우 크면 여러 가지 모델을 일정 시간 안에 훈련시킬 수 있도록 데이터를 샘플링하여 작은 훈련 세트를 만드는 것이 좋습니다(이렇게 하면 규모가 큰 신경망이나 랜덤 포레스트 같은 복잡한 모델은 만들기 어렵습니다).

• 여기에서도 가능한 한 최대로 이 단계들을 자동화합니다.)

**1. 여러 종류의 모델을 기본 매개변수를 사용해 신속하게 많이 훈련시켜봅니다(예를 들면 선형 모델, 나이브 베이지, SVM, 랜덤 포레스트, 신경망 등)**

**2. 성능을 측정하고 비교합니다.**

- 각 모델에서 N-겹 교차 검증을 사용해 N개 폴드의 성능에 대한 평균과 표준 편차를 계산합니다.

**3. 각 알고리즘에서 가장 두드러진 변수를 분석합니다.**

**4. 모델이 만드는 에러의 종류를 분석합니다.**

- 이 에러를 피하기 위해 사람이 사용하는 데이터는 무엇인가요?

**5. 간단한 특성 선택과 특성 공학 단계를 수행합니다.**

**6. 이전 다섯 단계를 한 번이나 두 번 빠르게 반복합니다.**

**7. 다른 종류 에러를 만드는 모델을 중심으로 가장 가능성이 높은 모델을 세 개에서 다섯 개 정도 추립니다.**

## B.6 시스템을 세밀하게 튜닝합니다.

(노트: • 이 단계에서는 가능한 한 많은 데이터를 사용하는 것이 좋습니다. 특히 세부 튜닝의 마지막 단계로 갈수록 그렇습니다.

• 언제나 그렇듯이 할 수 있다면 자동화합니다.)

**1. 교차 검증을 사용해 하이퍼파라미터를 정밀 튜닝합니다.**

- 하이퍼파라미터를 사용해 데이터 변환을 선택하세요. 특히 확신이 없는 경우 이렇게 해야 합니다(예를 들어 누락된 값을 0으로 채울 것인가 아니면 중간값으로 채울 것인가? 아니면 그 행을 버릴 것인가?).
- 탐색할 하이퍼파라미터의 값이 매우 적지 않다면 그리드 서치보다 랜덤 서치를 사용하세요. 훈련 시간이 오래 걸린다면 베이지안 최적화 방법을 사용하는 것이 좋습니다(예를 들면 가우시안 프로세스 사전 확률Gaussian process prior을 사용합니다).

**2. 앙상블 방법을 시도해보세요. 최고의 모델들을 연결하면 종종 개별 모델을 실행하는 것보다 더 성능이 높습니다.**

**3. 최종 모델에 확신이 선 후 일반화 오차를 추정하기 위해 테스트 세트에서 성능을 측정합니다.**

CAUTION 일반화 오차를 측정한 후에는 모델을 변경하지 마세요. 만약 그렇게 하면 테스트 세트에 과대적합되기 시작할 것입니다.

## B.7 솔루션을 출시합니다.

**1. 지금까지의 작업을 문서화합니다.**

**2. 멋진 발표 자료를 만듭니다.**

- 먼저 큰 그림을 부각시킵니다.

**3. 이 솔루션이 어떻게 비즈니스의 목표를 달성하는지 설명하세요.**

**4. 작업 과정에서 알게 된 흥미로운 점들을 잊지 말고 설명하세요.**

- 성공한 것과 그렇지 못한 것을 설명합니다.
- 우리가 세운 가정과 시스템의 제약을 나열합니다.

**5. 멋진 그래프나 기억하기 쉬운 문장으로 핵심 내용을 전달하세요 (예를 들면 중간 소득이주택 가격에 대한 가장 중요한 예측 변수입니다.).**

## B.8 시스템을 론칭합니다!

**1. 서비스에 투입하기 위해 솔루션을 준비합니다(실제 입력 데이터 연결, 단위 테스트 작성 등).**

**2. 시스템의 서비스 성능을 일정한 간격으로 확인하고 성능이 감소됐을 때 알림을 받기 위해모니터링 코드를 작성합니다.**

- 아주 느리게 감소되는 현상을 주의하세요. 데이터가 변화함에 따라 모델이 점차 구식이 되는 경향이 있습니다.
- 성능 측정에 사람의 개입이 필요할지 모릅니다(예를 들면 크라우드소싱crowdsourcing 서비스를 통해서)
- 입력 데이터의 품질도 모니터링합니다(예를 들어 오동작 센서가 무작위한 값을 보내거나, 다른 팀의 출력 품질이 나쁜 경우), 온라인 학습 시스템의 경우 특히 중요합니다.

**3. 정기적으로 새로운 데이터에서 모델을 다시 훈련시킵니다(가능한 한 자동화합니다).**