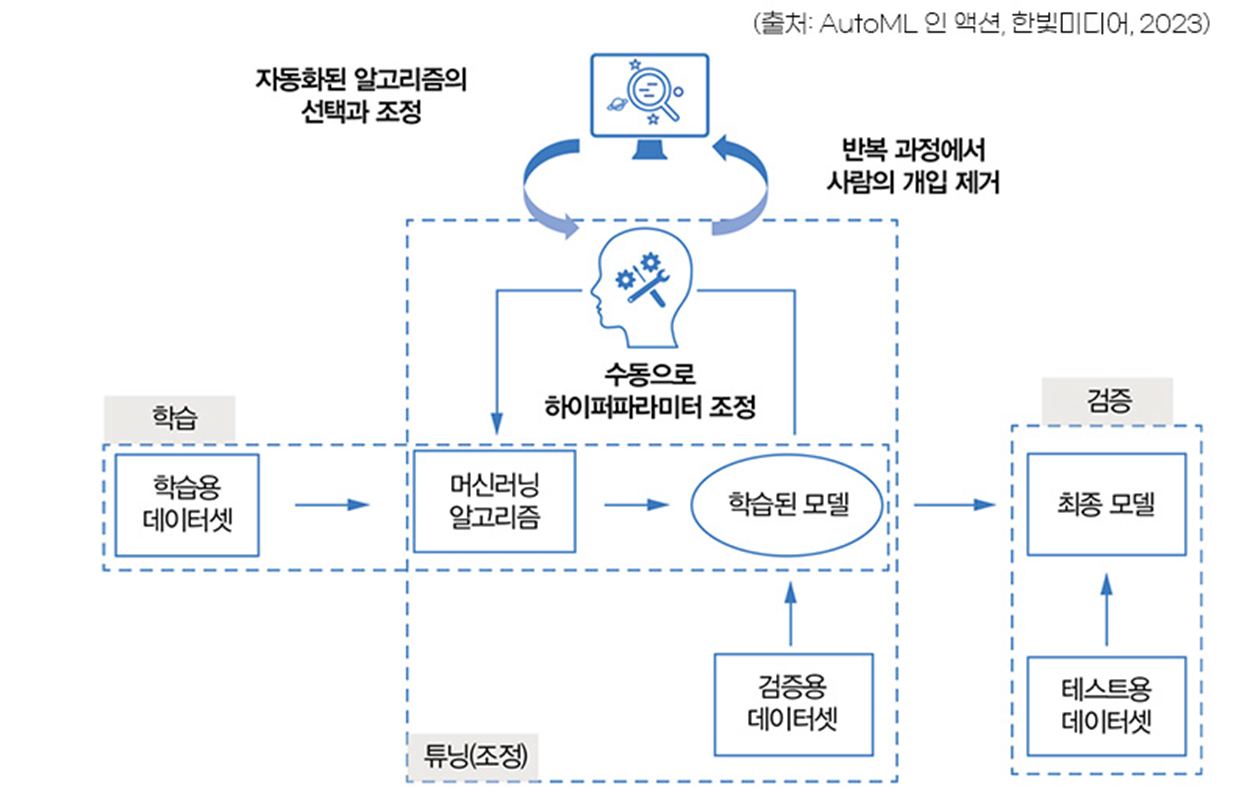
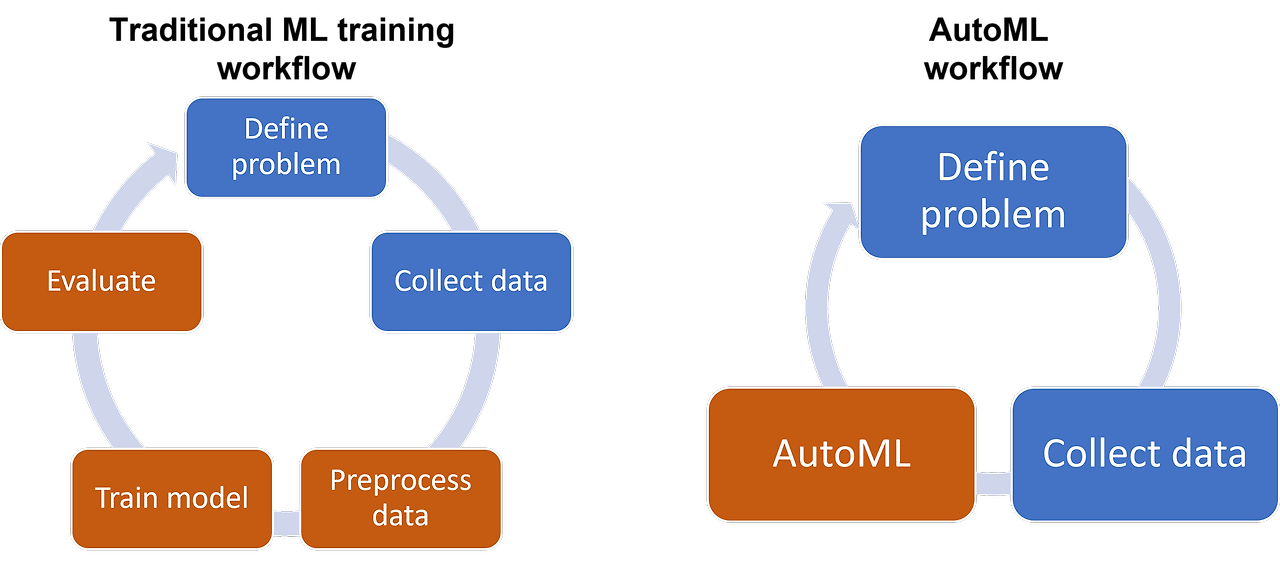
**AutoML**



1. AutoML(Automated Machine Learning)

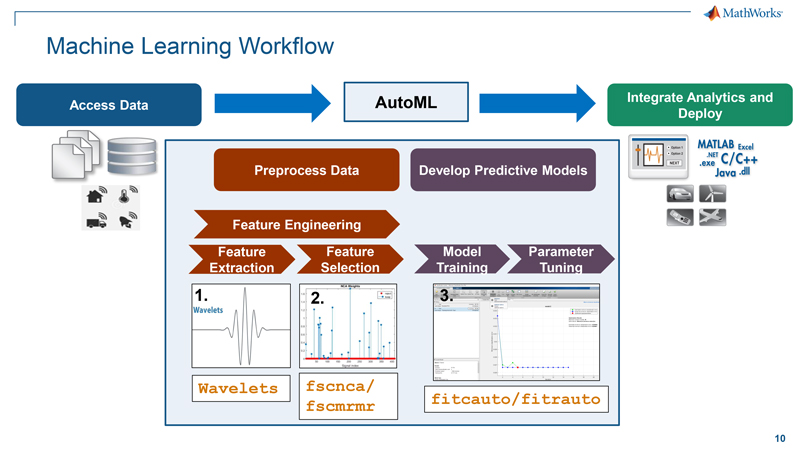


1) 시간 소모적이고 반복적인 기계 학습 모델 개발 작업을 자동화하는 프로세스

2) 머신러닝 모델의 학습과 배포 과정을 자동화하는 기술 또는 도구

 3) 기존의 머신러닝 전문가가 수행하던 영역 중 정형화된 과정을 자동화하여 업무 효율을 높여주고 비전문가들도 머신러닝 모델을 생성할 수 있도록 쉬운 인터페이스를 제공하는 것을 목표로 함

2.  AutoML의 장점



1) 모델의 품질을 유지하면서 확장성, 효율성, 생산성이 높은 ML모델을 구축할 수 있음

2) 머신러닝을 위한 고급 모델 구축을 자동화할 수 있음

    -> 지식과 스킬이 필요한 공정을 기계가 알아서 처리함

        (=전문적인 지식과 이해 없이도 필요에 맞게 고품질 모델을 학습시킬 수 있음)

    -> 누구나 쉽게 머신러닝을 활용하고 쉽고 빠르게 도입할 수 있도록 함

3) 모델 최적화 과정을 훨씬 더 효율적으로 발전시킬 수 있음(빠르게!)

## 3. AutoML의 소개를 위한 사전지식

1) (파이썬)라이브러리

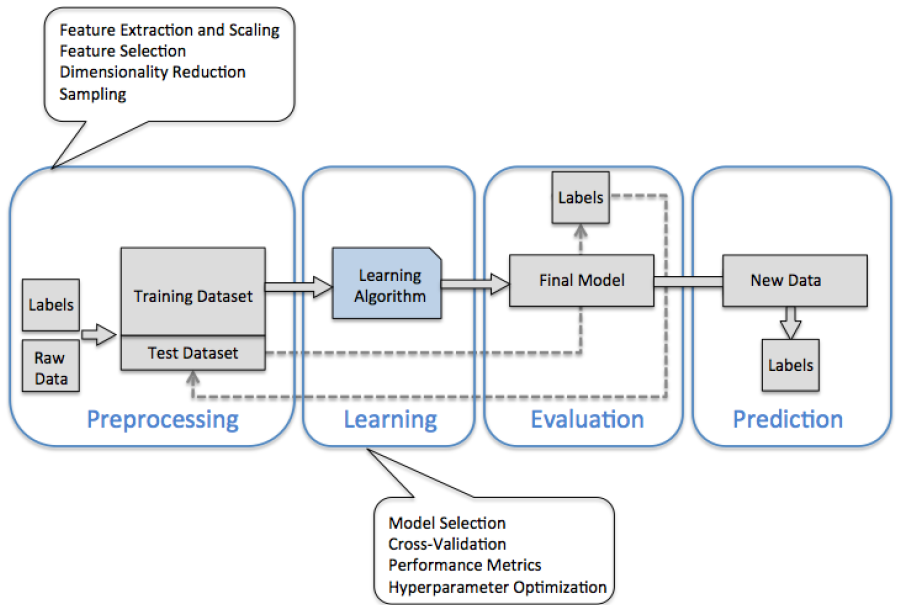
(1) 코드를 작성할 필요없이 많은 동작을 수행할 수 있게 해주는, 다양한 목적으로 사용되는 함수와 메서드의 집합

(2) 프로그래밍을 하면서 파이썬 라이브러리를 꺼내어 사용할 수 있음 (도서관(라이브러리)에서 책을 꺼내 보듯이)

(3) 이미 작성된 코드의 재사용이 가능하므로 프로그래밍 작업의 효율성과 속도를 증가시켜줄 수 있음

2) 머신러닝의 과정(프로세스)

: 데이터 전처리, 모델링, 후처리로 총 3단계로 이루어짐



(1) 데이터 전처리 : 학습 데이터 준비 작업을 함

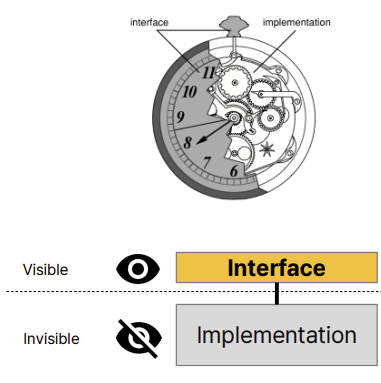
       ex) 데이터 수집, 데이터 클렌징, 결측값 처리, 분포 변환 등

(2) 모델링 : 여러 과정을 거쳐 최적의 결과를 도출함

* 변수 및 알고리즘 선택, 알고리즘 별 하이퍼파라미터 튜닝
* 후보 변수, 알고리즘, 하이퍼파라미터의 다양한 조합을 탐색하는 반복적인 계산 과정을 거침

(3) 후처리 : 모델의 활용성을 검증하고 실제 서비스에의 적용을 위한 작업을 함 모델의 활용성 검증 : 모델의 평가, 예측, 결과 해석, 모델 배포

3) 인터페이스



(1) 서로 다른 두 개의 시스템, 장치 사이에서(장치끼리) 정보나 신호를 주고받는(교환하는) 경우의 접점이나 경계면(일종의 공유 경계임)

(2) 사용자가 기기를 쉽게 동작시키는 데 도움을 주는 시스템

(3) 사물끼리, 사용자와 사물 간의 의사소통이 가능하게 하는 매개

(4) 기능, 서비스를 이용하기 위해서는 인터페이스에 대한 이해만 있으면 됨

(5) 자동차라면 핸들, 가속/브레이크 페달이 인터페이스

4) 오픈 소스

(1) Open Source Software 즉OSS를 의미함

(2) 소프트웨어 프로그램을 개발하는 과정에 필요한 소스 코드나 설계도를 누구나 접근해서 열람할 수 있도록(결국, 누구나 기술을 수정하고 공유할 수 있도록) 공개하는 것 또는 그런 모델

(3) 개방형 협업을 통해 개발 및 관리되는 소프트웨어

(4) 무료로, 누구나 원하는대로 사용, 검토, 변경, 재배포 가능함

(5) 오픈소스를 사용하는 이유와 이점

* 집단지성과 수천명이 주고 받는 피드백을 통해 빠르게 발전할 수 있음

(빠르고 다양한 개선 가능) -기업

* 조언이나 건의사항을 받아들여 프로젝트 진행시 유용함 -기업
* 공유를 목적으로 제작되었기 때문에, 개발비용이 낮음 -개발자

5) 커스터마이징

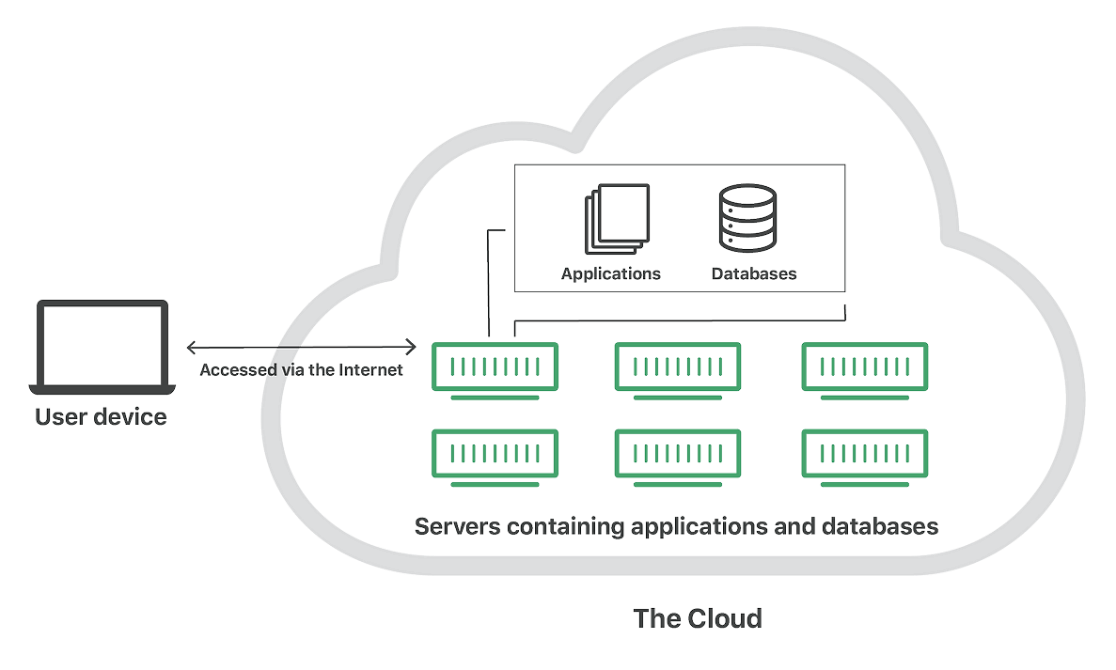
: solution 내부적으로 구현된 기능 중 일부를 사용자들의 요구사항을 반영할 수 있도록 수정하는 것

6) 컴퓨팅 자원

* 컴퓨터를 실행시키는 데 필요한 자원
* ex : 컴퓨터 네트워크, 데이터베이스, 서버, 스토리지, 애플리케이션

7) 클라우드(클라우드 환경) & 클라우드 컴퓨팅 & 클라우드 서비스

(1) 클라우드(클라우드 환경)

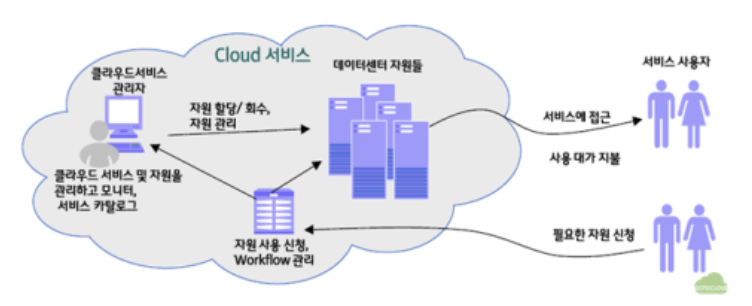


* 광대한 네트워크를 통하여 접근할 수 있는 가상화 된 서버와

서버에서 작동하는 프로그램과 데이터베이스를 제공하는 IT 환경

* 인터넷을 통해 접근할 수 있는 서버와 이러한 서버에서 작동하는 소프트웨어, 데이터베이스
* 필요한 컴퓨팅 자원을 인터넷으로 쉽게 이용할 수 있음
* =인터넷

(2) 클라우드 컴퓨팅



* 클라우드(인터넷)을 통해 가상화된 컴퓨터의 시스템 리소스(IT 리소스)를 요구하는

즉시 제공하는 것

* 인터넷 기반 컴퓨팅의 일종
* 정보를 자신의 컴퓨터가 아닌 클라우드(인터넷)에 연결된 다른 컴퓨터로 처리하는 것

(3) 클라우드 서비스



* 클라우드 컴퓨팅을 기반으로 제공되는 서비스
* 인터넷을 통해 컴퓨팅 자원, 데이터 저장, 소프트웨어, 플랫폼 및 기타 IT 관련 서비스를

원격으로 제공하는 것

* 클라우드(인터넷)에 존재하는 애플리케이션 및 인프라 리소스

(이용자는 인프라를 관리할 필요가 없어짐)

8) 파이프라인(=데이터파이프라인)

(1) 한 데이터 처리 단계의 출력이 다음 단계의 입력으로 이어지는 형태로 연결된 구조

(2) 여러 단계가 서로 동시에 또는 병렬적으로 수행될 수 있어서 효율성이 향상됨

9) 모델 최적화 과정

(1) 모델 최적화

: 더 나은 성능, 효율성, 형태, 또는 특정 요구사항을 충족시키도록 하는 것

(2) 모델 최적화 과정

* 모델을 최적화 시키는 과정
* 주어진 문제에 대해 최소한의 비용으로 최선의 목표를 찾는 과정
* 최적화 기법 중, 모델 경량화 같은 것들이 있음

10) Python API & UI

(1) Python API

* API : Application Programming Interface : 컴퓨터와 컴퓨터 프로그램(소프트웨어) 사이의 연결(연결하는 것)
* Python API : Python에서 API를 받아와서 이용하는 것/Python에서 제공하는 API

(2) UI

* UI : User Interface(사용자 인터페이스)
* 사람들이 컴퓨터와 상호 작용하는 시스템
* 사용자와 사물, 시스템, 기계, 컴퓨터 프로그램 등 사이에서 의사소통을 할 수 있도록 일시적 또는 영구적인 접근을 목적으로 만들어진 물리적, 가상적 매개체

4. AutoML의 종류(AutoML을 실행하는 방식의 종류)

1) OSS(Open Source Software) 방식

(1) 오픈 소스 기반의 라이브러리 형태로 제공되는 AutoML 기능

(2) OSS 방식을 사용하는 AutoML 라이브러리의 종류

* Scikit-learn 기반 : Auto-sklearn, TPOP
* Keras 기반 : AutoKeras
* Tensorflow 기반 : AdaNet

: Scikit-learn 기반의 라이브러리는 구조화된 데이터에 적합하며 기존 Scikit-learn의 전처리 방식을 자동화하여 적용이 가능함

: Keras 기반의 라이브러리와 Tensorflow 기반의 라이브러리는 딥러닝 모델(텍스트, 이미지 처리 등)에 적합하며 효율적인 신경망 구조 탐색이 가능함

(3) OSS 방식의 장단점

* 장점 : 기존의 코드에 AutoML 라이브러리를 호출하여 사용할 수 있기 때문에 커스터마이징이 자유로움
* 단점 1 : 높은 컴퓨팅 자원 활용이 가능한 환경을 갖춰야함(환경적으로 제약이나 한계가 있음)
* 단점 2 : 후처리 단계(모델 평가, 배포, 모니터링 등)에 대한 기능 지원이 미흡함

2) Cloud Provider Solution 방식

(1) 주요 클라우드 서비스에서 제공하는 클라우드 환경에서 이용 가능한 AutoML solution

(2) Cloud Provider Solution을 사용하는 AutoML 라이브러리의 종류

* Google : Cloud AutoML
* Amazon : Sagemaker Autopilot
* Microsoft : Azure

(3) 코드 작성이 필요없는 UI 방식과 Python API 방식 둘 다를 제공함

(4) Cloud Provider Solution 방식의 장단점

* 장점 1 : 클라우드 환경에서 제공되므로, 필요한만큼의 자원 할당이 가능하며 자원 관리 서비스가 제공됨
* 장점 2 : 데이터 전처리, 모델링, 후처리(결과 평가, 모델 배포)까지의 기능을 제공하므로 전체 프로세스에 대한 구현이 가능함
* 단점 : 사용자가 구현된 시스템 내부를 투명하게 파악하기는 어려움(클라우드 서비스에서 제공한 것이기 때문임)

3) Enterprise Solution 방식

(1) AutoML 서비스 제공을 목적으로 만들어진 전문 AutoML platform(solution)

(2) Enterprise solution 방식을 사용하는 AutoML 라이브러리의 종류

* DataRobot
* H2O

(3) 원하는 곳에 설치 가능하므로 민감한 데이터를 다루는 경우 적합함

(4) Enterprise solution 방식의 장단점

* 장점 : AutoML에 특화된 기능과 각 프로세스에 커스터마이징이 가능함
* 단점 : 초기의 라이선스 구매 비용이 듦

5. 많이 사용되는 AutoML 라이브러리

1) LightAutoML

(1) 자동화된 기계 학습을 목표로 하는 오픈 소스 파이썬 라이브러리

(2) 데이터의 형태 : 텍스트

(3) 다양한 작업에 대해 가볍고 효율적으로 설계함

2) H2O AutoML

(1) 알고리즘 선택, features 생성, 하이퍼파라미터 튜닝 및 반복 모델링 을 자동화하는 기계 학습 프로젝트에서 주로 사용됨

(2) 기계 학습 프로젝트가 오류없이 효율적으로 ML 모델을 훈련하고 평가하는 데 도움을 줌

(3) 프로젝트의 성능 향상을 위한 기계 학습 전문 지식이 덜 요구됨(이건 AutoML의 전반적인 특성임)

3) TPOT

(1) Scikit-learn 데이터의 preparation 및 기계 학습 모델이 있는 오픈 소스 라이브러리

(2) genetic programming으로 머신러닝 파이프라인을 최적화하는 파이썬 AutoML 도구임

(3) 예측 모델링 작업을 위한 고품질의 기계 학습 모델을 자동으로 발견하기 위해 사용됨

(4) 수천 개의 가능한 파이프라인 중에서 가장 적합한 것을 사용하여 규칙적이고 지루한 작업을 자동화하는 데 도움이 됨

4) Pycaret

(1) 머신러닝 workflow를 자동화하는 오픈 소스 라이브러리

(2) (classification, regression, clustering 등의) 작업(task)에서 사용하는

여러 모델들을 동일한 환경에서 한 번에 한 줄의 코드로 실행할 수 있도록 자동화한 라이브러리

(3) 여러 모델의 비교가 가능함

(4) 각 모델 별로 튜닝할 수 있음

6. 머신러닝 모델 개발에서의 AutoML(라이브러리) 사용

여러 환경, 수준, 비용에 따라 어떤 AutoML을 사용할 지를 선택해야함

각 라이브러리마다도 장단점이 있으므로 필요한 라이브러리를 적절히 골라야함

1) 개발자의 수준에 따라

(1) 비전문가, 미숙한 개발자 : 기업의 platform, solution 사용(UI 방식)

(2) 능숙한 개발자

(전처리, 모델링, 후처리 등의 전반적인 머신러닝 프로세스에 능숙) :

오픈소스 라이브러리를 활용하여 초기 모델 만들기

-> 커스터마이징을 통해 고도화해나가기

2) 사용자의 데이터 및 개발 환경에 따라

3) 예산이나 비용에 따라

**참고**

* <https://mz-moonzoo.tistory.com/5>
* <https://data-minggeul.tistory.com/12>