

연구인력 현장지원 실적보고서(완료)

----- 연구자 비밀유지 의무 공지 -----

**협약서 7조(비밀유지 의무)** 연구자는 지원기업에서 업무수행 중 지득한 비밀을 지원기업의 승인 없이 제3자에게 누설하여서는 아니 되며, 누설 시 지원기업은 연구자에게 손해배상을 청구할 수 있다.

-----

<b>지원 기업명</b>	HNCN ( 대표: 이영우 )
<b>연구자(지원인력) (직급 및 성명)</b>	선임연구원 박 정 우
<b>지원 기간</b>	2024. 9. 1. ~ 2025. 8. 31. ( 12 개월)
<b>지원 유형</b>	R&BD기획(    ), 기술개발( √ ), 기술지도/자문(    ), 사업컨설팅(    ), 기술마케팅(    ), 기타(내용:                      )
<b>지원 목표 (기업요청 내용)</b>	○ 스마트공장 분야 사업 고도화를 위한 산업현장에서의 인공지능 적용분야 발굴 및 기술 개발을 추진하기 위해 필요한 데이터셋 구축 및 AI 솔루션 개발을 기술지원해 줄 인공 지능 전문가를 필요로 함 ○ CNC머신의 공정 데이터를 머신러닝으로 분석하여 설비 노후화에 따른 가공 불량률의 발생 여부를 실시간 예측하고 피쳐 중요도를 추출함으로써 불량률 예측시 신속한 피쳐별 실시간 점검 및 즉각 조치를 가능하게 함
<b>수행 업무</b>	○ 기업현장의 인공지능 적용분야 발굴을 위한 현장 실무회의: 분야별 주요 인공지능 기술 및 적용사례 소개 ○ 제조업 내 CNC머신 업종의 데이터 전처리 기법을 우선 확보한 다음에, 이를 처리할 머신러닝 기법의 적용을 통해 공정최적화 기법과 중요 공정변수의 추출 방법을 도출함 ○ 기업의 주요 사업수단인 로봇 관련분야 진출을 위한 PoC 개발 - 강화학습 이론, PPO 알고리즘 스터디 및 PoC 아이템 선정 - 강화학습 알고리즘 개발을 위한 클라우드 개발 인프라(AWS) 구축
<b>지원성과</b>	○ 본 데이터 분석, 전처리 및 AI 모델을 이용하여 CNC 공작기계에서 발생하는 불량 가공품의 실시간 감지 및 피쳐 중요도에 따른 신속한 피쳐별 점검을 통해 지속적인 불량품 양산을 사전에 예방함으로써 기업의 손실 감소 및 영업이익 증대에 기여함

	○ CNC가공기계는 컴퓨터를 통해 기계의 움직임, 속도, 회전수 등을 지정하기 위해 G-code라는 프로그래밍 언어를 사용하여 제어되고 CNC선반 및 CNC밀링의 데이터 형태는 거의 표준화되어 있으므로, 본 사업수행 결과물을 다수의 유사한 CNC 선반 및 CNC 밀링 설비에 폭넓게 활용 가능함
--	--

연구인력 현장지원 실적보고서를 첨부와 같이 제출합니다.

2025 . 09. 03.

보고자 성명: 박 정 우 (인)

(기업체 확인) 소속: AI 연구실                      직위/직급: 전무  
성명: 정 태 승 (인)

(첨부) 연구인력 현장지원 실적보고서(상세)

# 연구인력 현장지원 실적보고서(상세)

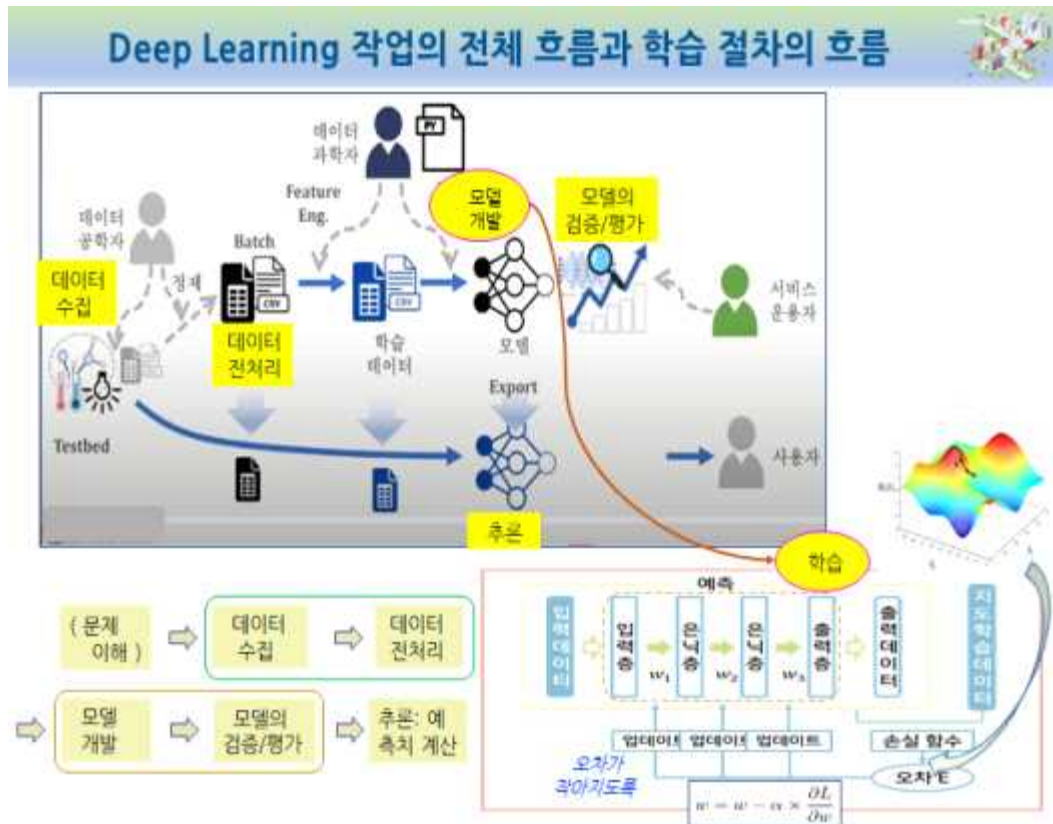
## 1.수행업무 1: 기업현장의 인공지능 적용분야 발굴을 위한 현장 실무회의 - 분야별 주요 인공지능 기술 및 적용사례 소개

- 주요 인공지능의 종류를 개관하고, 일반적인 딥러닝 작업의 전체 흐름과 학습절차를 리뷰함
- 인공지능의 적용 사례1로서 예측 태스크를 소개함
- 인공지능의 적용 사례2로서 수명예측 태스크를 소개함
- 인공지능의 적용 사례3으로서 이상탐지 태스크를 소개함

### □ 결과:

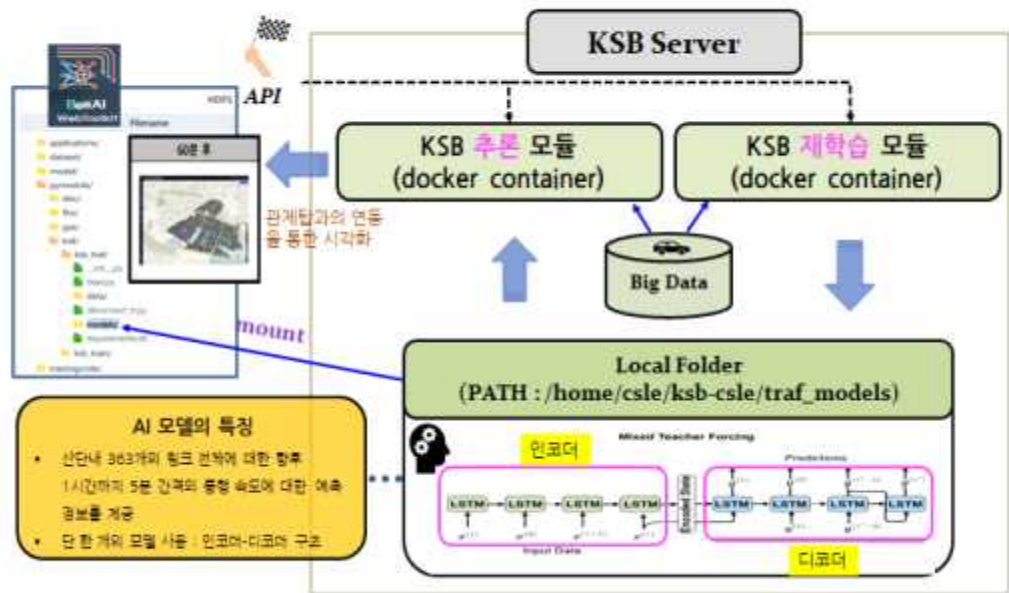
- 주요 인공지능의 종류 및 일반적인 딥러닝 작업의 전체 흐름과 학습절차

### 지원 내용 및 결과



- 인공지능의 적용 사례1 - 예측 태스크

## 적용 사례 1a: AI플랫폼 서버의 구성 및 AI 교통예측 서비스의 개요

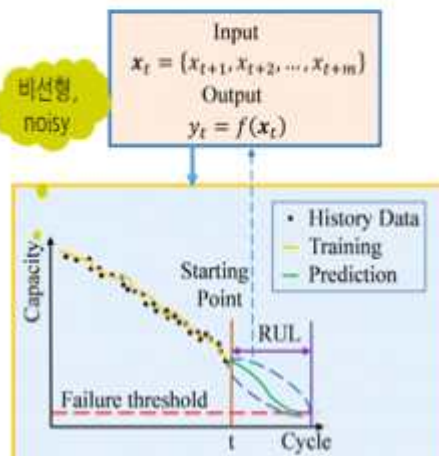
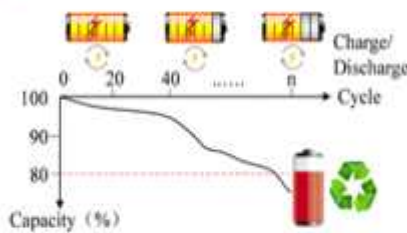


- 인공지능의 적용 사례2 - 수명 예측 태스크를 소개

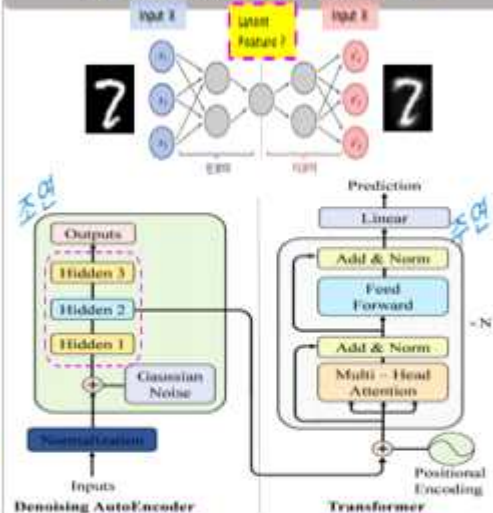
## 적용 사례 2a -인공지능에 의한 수명 예측

### 개념 / 아이디어

잔여유효수명(RUL) 예측 (건전성 평가) 기술이란? = 회귀 문제



### 구현(DeTransformer)의 개요



노이즈로 가득한 배터이용량 데이터 → 오토인코더

→ Transformer 모델

- 예측 Task를 방해하는 노이즈 문제를 완화하기 위해 Raw Data를 Denoising Autoencoder로 우선 처리
- 시간적 정보와 유용한 특징을 학습하기 위해 트랜스포머로 보내서 일반적인 RUL 예측 Task 수행 → MHA에 학습을 가속화

노이즈 제거 - 예측 Task를 연결하기 위해 목표 함수에 반영

- 목표 함수 = DAE Loss + Prediction Loss → 동시에 최적화

● 인공지능의 적용 사례3 - 이상 탐지 태스크를 소개

**적용 사례 3 -인공지능에 의한 고장진단 / 이상탐지**

### 개념 / 아이디어

● 인공지능에 의한 고장진단 기술

1단계: Autoencoder 학습

2단계: 적대적 학습

### 모델의 특징

● 기존 Autoencoder 기반 이상탐지 모델의 구조를 개선

- 비지도학습 방식의 Autoencoder 기반 이상탐지 모델은 정상데이터를 잘 복원하도록 학습되기 때문에 **이상치**가 정상데이터와 유사한 경우에 재구성 오류가 작아 이상치로 탐지되지 않는 문제들 -> GAN에 의한 **적대적학습** 개념의 적용을 통해 재구성 오류를 증폭시킴으로써 보완
- 1단계의 Autoencoder 학습에서는 각 Encoder가 입력된 Latent space로 압축된 후, Decoder를 통해 **일차와 동일하도록 복원**하는 Autoencoder를 학습

● 기존 GAN 기반 이상탐지 모델의 한계를 개선

- GAN 기반 이상탐지 모델의 학습 불안정 / Non-convergence 및 Mode Collapse의 문제를 개선
- 2단계의 적대적 학습에서는 **두개의 Autoencoder를 각각 GAN구조의 Generator 및 Discriminator로 동작**시켜서 적대적 학습을 진행시킴
- Autoencoder 구조와 적대적학습 개념의 융합을 통해 학습과정의 고수확률과 동시에, Autoencoder가 **정상데이터와 유사한 이상치**의 탐지도 가능하게 됨
- GAN 기반 모델임에도 불구하고 Autoencoder 구조를 통해 **안정적인 학습**이 가능해지는 효과를 기대

24

2. 수행업무 2:

제조업 내 CNC 머신 업종의 데이터 전처리 기법을 우선 확보한 다음에, 머신러닝 기법의 적용을 통해 공정최적화 기법과 중요 공정변수의 추출 방법을 도출함

- 개발환경으로는 중소벤처기업부에서 최근 NHN클라우드와 손잡고 구축한 클라우드 기반의 개발환경인 인공지능 중소·벤처 제조플랫폼 'KAMP'(Korea AI Manufacturing Platform)를 이용하였음
- CNC 머신의 공정데이터 분석을 위한 데이터로는 동 사이트의 오픈소스를 사용하였고, X,Y,Z축 및 스핀들 방향의 위치,속도,가속도 등의 피처(변수) 총 48가지를 수집하였으며, 데이터 형태는 CSV 파일임
- 수집된 데이터 갯수는 96,169개이고, 데이터셋 사이즈는 14.62 MB임
- 분석에 적용된 알고리즘으로는 상호 비교에 의한 객관성 확보를 위해 LightGBM 모델과 랜덤포레스트 모델의 2종을 적용하였음
- LightGBM은 여러 개의 약한 학습기를 결합하여 강한 학습기를 만드는 기법인 Gradient Boosting 알고리즘을 더 빠르고 효율적으로 실행되도록 MS사가 발전시킨 Gradient Boosting 프레임워크로서, Leaf-wise(잎 방향의) 트리 성장방식을 사용하며 XGBoost에 비해 훈련시간이 짧고 성능도 좋음
- 랜덤 포레스트는 의사결정나무를 배경 방식으로 만든 머신러닝 알고리즘의 일종으로 앙상블 학습 방법론을 사용한 결정 트리의 확장 버전으로서, 부트스트랩을 통해 다양한 서브 데이터셋을 생성하여 다수의 결정트리를 학습시킨 다음에 그 결과를 집계해 최종 예측결과를 도출하는 방식으로 작동함



- 랜덤 포레스트는 단일 의사결정나무가 가질 수 있는 과적합 문제를 해결할 수 있으며, 보다 강력하고 견고한 예측 성능을 기대할 수 있는 알고리즘임
- 상기 2종의 모델을 각각 학습시킨 다음에 각 모델을 사용하여 예측을 수행하고 그 성능을 평가한 결과, LightGBM 모델의 경우에 학습 정확도는 97%이나 테스트 정확도는 95%로 학습 성능과의 차이가 미미함을 확인함
- 이와 달리, 랜덤포레스트 모델의 경우에 학습 정확도는 100%이나 테스트 정확도는 86%로 학습 성능과의 차이가 상당하였고, 이에 과적합이 의심되어 Hyperparamter 튜닝으로 성능 개선을 시도하였음
- 이를 통해 업그레이드된 새로운 모델을 생성한 다음에 재학습시켜서 재예측 및 재평가를 수행한 결과, 학습 정확도는 94%이나 테스트 정확도는 90%로 학습 성능과의 차이가 상당히 줄어들었고 성능이 개선되었음
- 상기 2종의 모델 각각에 대해 피쳐 중요도 분석에 의해 중요변수를 파악하는 과정을 통해 불량 여부의 최종 판단에 영향을 미치는 피쳐별 기여도를 상위 10위까지 순위 도출하였으며, 이 결과에 의거하여 현업의 공정 실무자는 가공 조건을 조정함으로써 불량률을 최소화할 수 있게 됨
- LightGBM 모델에 의한 피쳐 중요도 분석 결과, 순서대로 'S\_SetPosition', 'S\_ActualPosition', 'M\_CURRENT\_FEEDRATE'을 상위 3위까지의 피쳐로 도출함
- 랜덤포레스트 모델에 의한 피쳐 중요도 분석 결과, 순서대로 'S\_ActualPosition', 'M\_sequence\_number', 'M\_CURRENT\_FEEDRATE'을 상위 3위까지의 피쳐로 도출함
- 최종적으로 모델링1과 모델링2에 의한 피쳐 중요도 분석 결과를 상위 3위까지의 피쳐로 종합해 보면, 두 알고리즘에 공통적으로 'S\_ActualPosition',과 'M\_CURRENT\_FEEDRATE'의 기여도가 순서대로 가장 큰 것으로 판단하는 것이 타당하다는 결론을 도출함

#### □ 결과:

- 데이터 EDA 및 전처리를 위한 주요코드

```
# Label 0(양해) / 1(불량해) --> data01 / data02+data03

# 전체 데이터 중 평가(테스트) 데이터셋으로 할당할 비율을 설정한다.
# 평가(테스트) 데이터를 (1-portion)의 비율만큼 미리 떼어놓기 위한 작업이다.
portion = 0.95

# 'data_pass'에서 평가를 데이터를 제외한 나머지 부분을 학습 데이터로 할당한다.
# 이때 'portion' 비율만큼 데이터를 학습용으로 사용한다.
data01 = data_pass[0:int(portion*npass),:]

# 'data_pass_half'와 모든 데이터를 학습 데이터로 사용한다.
data02 = data_pass_half[0:npass_half,:]

# 'data_fail'에서 일부 데이터를 학습 데이터로 할당한다.
# 여기서는 'portion' 비율만큼 'data_fail' 데이터에서 'npass_half'를 한 만큼을 사용한다.
data03 = data_fail[0:int(portion*npass)-npass_half,i]

# 학습 데이터셋의 생성
data_trn = np.concatenate((data01,data02),axis=0); # 'data01'과 'data02'를 합친다.
data_trn = np.concatenate((data_trn,data03),axis=0);
# 위에서 합친 데이터에 'data03'을 추가로 합쳐 최종 학습 데이터셋을 생성한다.

# 평가 데이터셋의 생성
# 'data_pass'에서 'portion' 비율 이후의 데이터를 평가를 데이터로 할당한다.
data_to = data_pass[int(portion*npass):npass,:];

# 'data_fail'에서 특정 범위와 데이터를 평가를 데이터로 할당한다.
# 여기에서는, 'data_fail'의 'portion' 비율 이후 데이터 중 일부를 사용하며
# 구체적으로는, 열 구간 지정시 [시점 - npass_half, 종결 - npass_half]로 지정해 주었다.
data_tx = data_fail[int(portion*npass) - npass_half : npass - npass_half, :];

# 최종적으로 'data_to'와 'data_tx'를 합쳐 평가 데이터셋을 생성한다.
data_test = np.concatenate((data_to,data_tx),axis=0);
```

- LightGBM 모델의 분류 성능

```
[54]: # 분류 결과를 평가
from sklearn.metrics import classification_report

#Y_pred_train = lgbm.predict(X_train) # 랜덤포레스트와 달리, 예측결과를 출력하는 이진화 추가적인 코드 필요
for i in range(0, len(Y_pred_train)): # 결과값을 출력
    if Y_pred_train[i] > 0.5: # Threshold를 0.5로 설정
        Y_pred_train[i] = 1
    else:
        Y_pred_train[i] = 0

#Y_pred_test = lgbm.predict(X_test)
for i in range(0, len(Y_pred_test)):
    if Y_pred_test[i] > 0.5: # Threshold를 0.5로 설정
        Y_pred_test[i] = 1
    else:
        Y_pred_test[i] = 0

print(classification_report(Y_train, Y_pred_train)) # 학습 데이터에 대한 분류결과를 평가
print(classification_report(Y_test, Y_pred_test)) # 테스트 데이터에 대한 분류결과를 평가
```

	precision	recall	f1-score	support
0	0.96	0.98	0.97	14212
1	0.98	0.96	0.97	14212
accuracy			0.97	28424
macro avg	0.97	0.97	0.97	28424
weighted avg	0.97	0.97	0.97	28424

	precision	recall	f1-score	support
0	1.00	0.91	0.95	748
1	0.92	1.00	0.96	748
accuracy			0.95	1496
macro avg	0.96	0.95	0.95	1496
weighted avg	0.96	0.95	0.95	1496

- 랜덤포레스트 모델의 분류 성능

```
[58]: # 랜덤포레스트 모델의 하이퍼파라미터를 사용하여 새로운 모델을 생성 및 학습시킨다.
maxDep, nEST = 10, 300

rfc = RandomForestClassifier(n_estimators=nEST, max_depth=maxDep, random_state = 12345)
rfc.fit(X_train, Y_train)

# 다시 한번 훈련 및 테스트 데이터에 대한 예측을 수행하고 성능을 평가
Y_pred_train = rfc.predict(X_train)
Y_pred_test = rfc.predict(X_test)

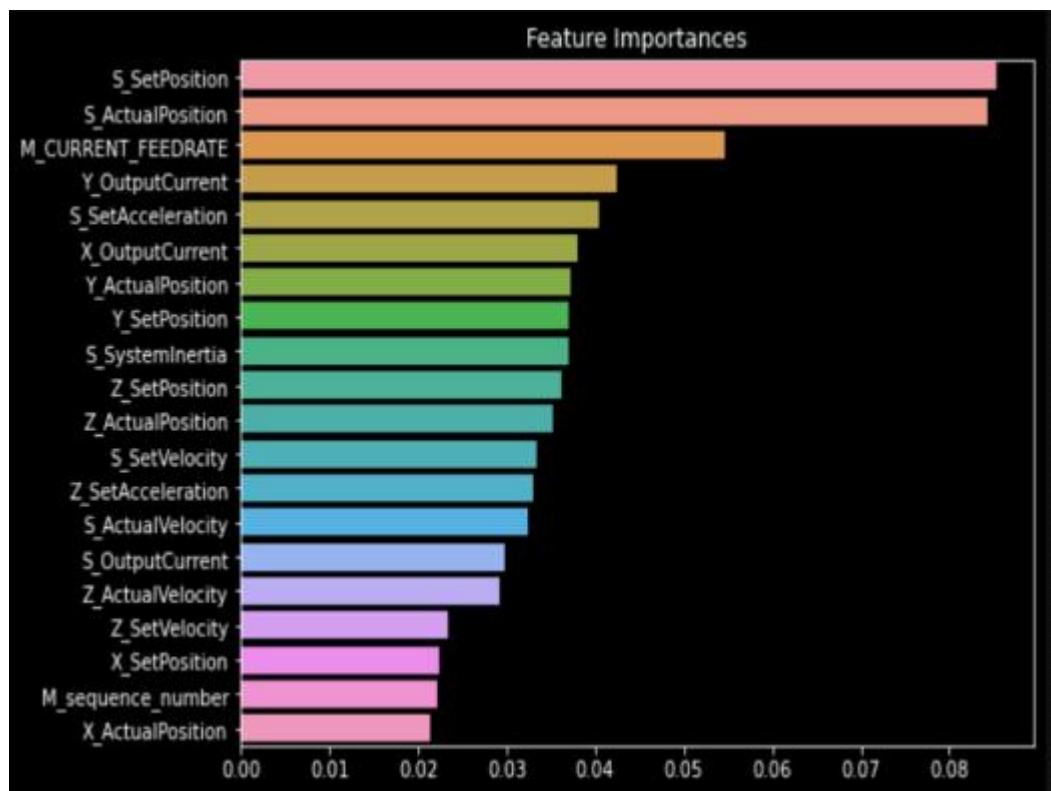
print(classification_report(Y_train, Y_pred_train))
print(classification_report(Y_test, Y_pred_test))
```

	precision	recall	f1-score	support
0	0.90	0.99	0.94	14212
1	0.99	0.89	0.94	14212
accuracy			0.94	28424
macro avg	0.94	0.94	0.94	28424
weighted avg	0.94	0.94	0.94	28424

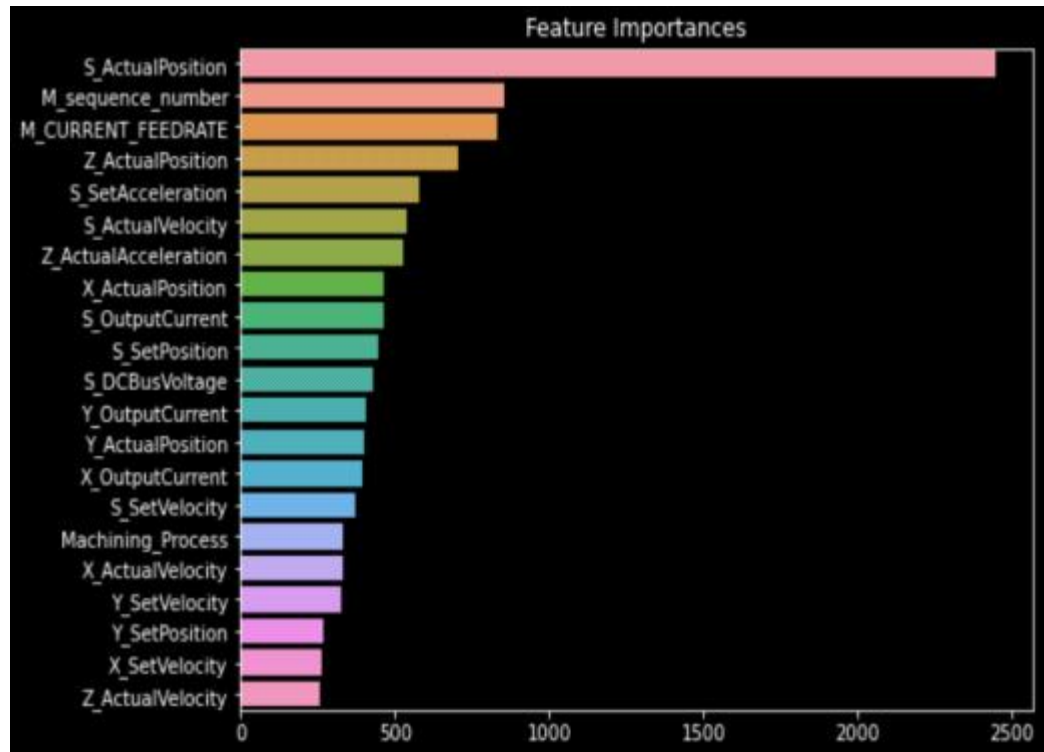
	precision	recall	f1-score	support
0	0.84	0.98	0.90	748
1	0.97	0.81	0.89	748
accuracy			0.90	1496
macro avg	0.91	0.90	0.90	1496
weighted avg	0.91	0.90	0.90	1496

- LightGBM 모델에 의한 피쳐 중요도 분석 결과



- 랜덤포레스트 모델에 의한 피쳐 중요도 분석 결과





### 3. 수행업무 3: 로봇 관련분야 진출을 위한 PoC 개발- 강화학습 이론, PPO

알고리즘 스터디 및 PoC 아이템 선정

- 강화학습 알고리즘 개발을 위한 클라우드 개발 인프라(AWS) 구축

- 고전적 강화학습의 기본이론 파악 : 환경, 에이전트, 상태, 행동, 보상, 전이확률, 정책의 개념 및 Markov 의사결정 과정 이해
- 벨만 기대방정식 및 벨만 최적방정식의 이해 및 분석
- Markov 의사결정 과정을 알 경우의 플래닝(정책 개선) 알고리즘 이해
- 인공지능 및 강화학습을 위한 개발 인프라 구축  
: 클라우드상에 아마존 가상머신(EC2 서버) 및 AWS 환경 구축  
-> Sagemaker Studio Lab을 이용한 서버 구축

#### □ 결과:

- 세미나 발표자료 중 <마르코프 의사결정 과정>

## 마르코프 의사결정 과정의 개요

Markov Decision Process

$$MDP \equiv (S, A, P, R, \gamma)$$

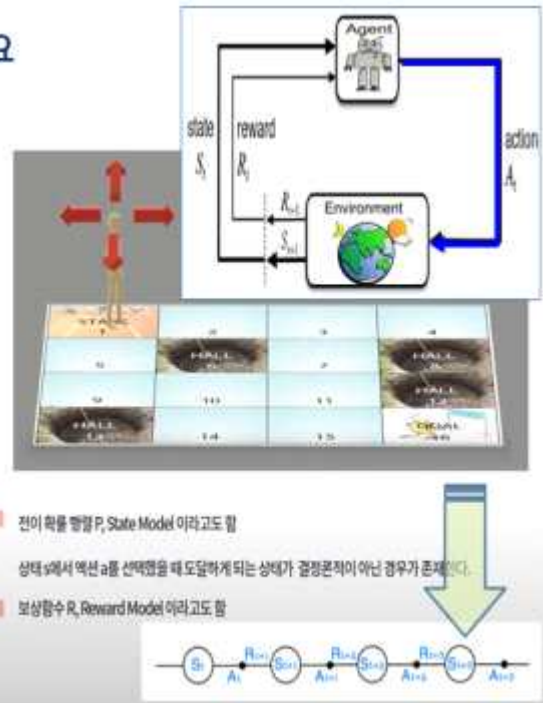
$$S = \{s_1, s_2, s_3, \dots, s_{15}, s_{16}\}$$

$$A = \{left, down, up, right\}$$

$$P_{ss'}^a = P[S_{t+1} = s' | S_t = s, A_t = a] \quad \leftarrow \text{전이 확률 } P, \text{ State Model 이라고도 함}$$

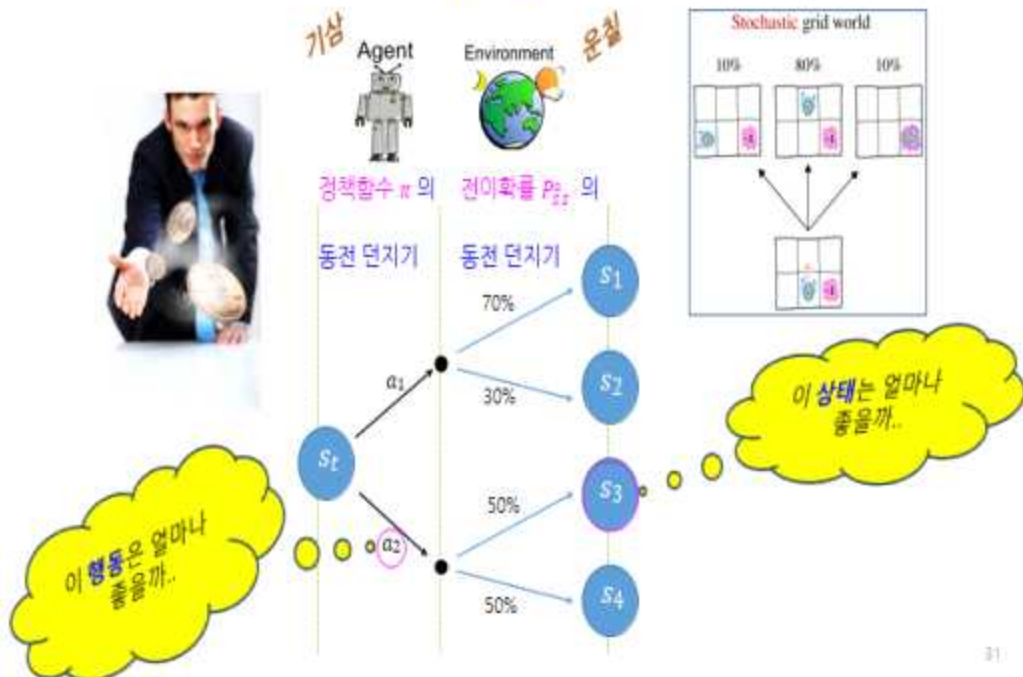
$$R_s^a = R[R_{t+1} | S_t = s, A_t = a] \quad \leftarrow \text{보상 함수 } R, \text{ Reward Model 이라고도 함}$$

$$\gamma = 0.99$$



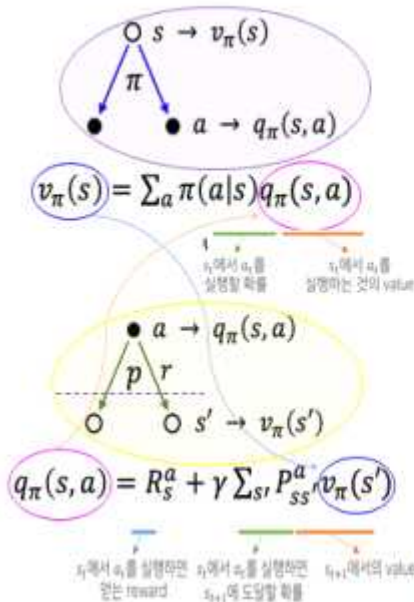
- 세미나 발표자료 중 <상태 전이를 위한 두 번의 샘플링>

## 상태 전이를 위한 두 번의 동전 던지기(샘플링)

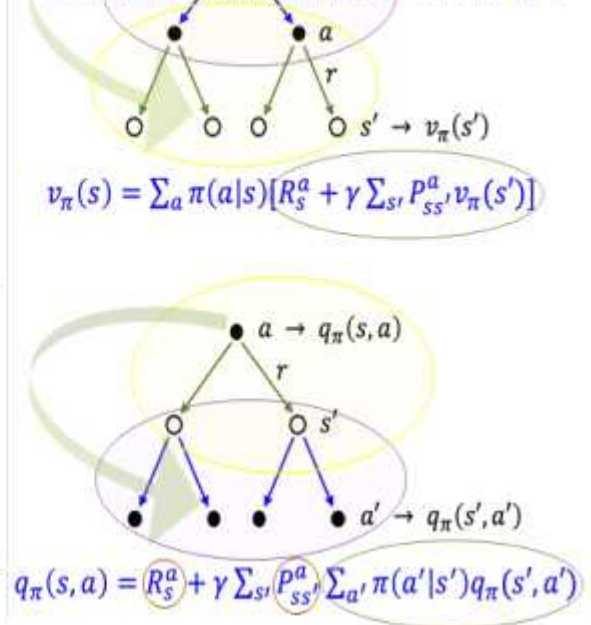


- 세미나 발표자료 중 <상태 가치함수-행동 가치함수 관계>

### 상태 가치함수 - 행동 가치함수 관계

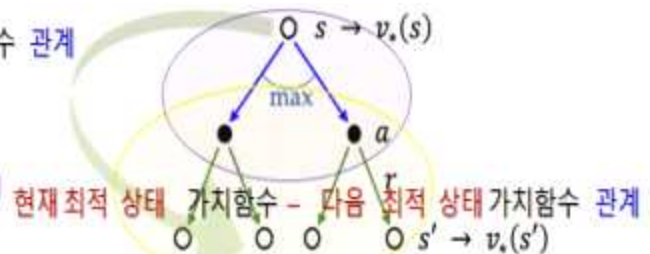
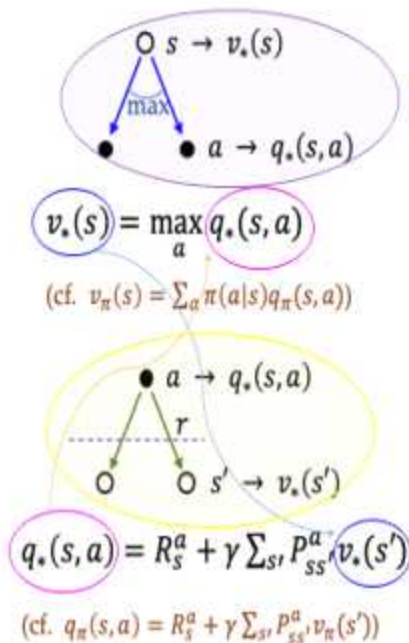


### 현재 상태 가치함수 - 다음 상태 가치함수 관계



- 세미나 발표자료 중 <최적 상태 가치함수 - 최적 행동 가치함수 관계>

### 최적 상태 가치함수 - 최적 행동 가치함수 관계



### 현재 최적 상태 가치함수 - 다음 최적 상태 가치함수 관계

$$v_*(s) = \max_a [R_s^a + \gamma \sum_{s'} P_{ss'}^a v_*(s')]$$

(cf.  $v_\pi(s) = \sum_a \pi(a|s) [R_s^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s')]$ )

$$q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \max_{a'} q_*(s', a')$$

(cf.  $q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \sum_{a'} \pi(a'|s') q_\pi(s', a')$ )

## ● 아마존 가상머신(EC2 서버)의 구축 현황

The screenshot displays the AWS Management Console for an EC2 instance. The instance is named 'i-06cf9b8ea8857843 (R1server6)' and is in the 'Running' state. Key details include:

- Instance ID:** i-06cf9b8ea8857843 (R1server6)
- IP 주소:** -
- 프라이빗 IP 주소:** 172.31.13.179
- 프라이빗 IP DNS:** ec2-3-35-92-92.ap-northeast-2.compute.amazonaws.com
- AMI ID:** ami-0a92388a6171e473
- AMI 이름:** Ubuntu/UNIX
- 인스턴스 유형:** t2.micro
- 인스턴스 프로파일:** aws-ec2-northeast-2-975049955477
- 인스턴스 상태:** Running
- 인스턴스 유형:** t2.micro
- 인스턴스 프로파일:** aws-ec2-northeast-2-975049955477
- 인스턴스 유형:** t2.micro
- 인스턴스 프로파일:** aws-ec2-northeast-2-975049955477

## ● 아마존 Sagemaker Studio Lab을 이용하여 구축된 AWS 서버 환경

The screenshot shows the Amazon SageMaker Studio Lab interface. The Jupyter Notebook displays a candle image and text explaining the environment setup:

We chose the MlpPolicy because the observation of the CartPole task is a feature vector, not images.

The type of action to use (discrete/continuous) will be automatically deduced from the environment action space

Here we are using the *Proximal Policy Optimization* algorithm, which is an Actor-Critic method. It uses a value function to improve the policy gradient descent (by reducing the variance).

It combines ideas from *A2C* (having multiple workers and using an entropy bonus for exploration) and *TRPO* (it uses a trust region to improve stability and avoid catastrophic drops in performance).

TRPO is an on-policy algorithm, which means that the trajectories used to update the networks must be collected using the latest policy. It is usually less sample efficient than off-policy algorithms like *SAC* or *TD3*, but is much faster regarding wall clock time.

```
[1]: env = gym.make("CartPole-v1")
      model = PPO2(policy, env, verbose=1)
```

We create a helper function to evaluate the agent

<p><b>지원성과/ 기대효과</b></p>	<ul style="list-style-type: none"> <li>○ 지원에 따른 예상/기대효과에 대해 기술 <ul style="list-style-type: none"> <li>- CNC 가공기계는 컴퓨터를 통해 기계의 움직임, 속도, 회전수 등을 지정하기 위해 G-code라는 프로그래밍 언어를 사용하여 제어되고 CNC 선반 및 CNC 밀링의 데이터 형태는 기본적인 수준에서는 표준화되어 있으므로, 본 사업수행 결과물을 정밀기계공업 분야 제조기업 내 다수의 유사한 CNC 선반 및 CNC 밀링 설비에 폭넓게 활용 가능함</li> <li>- 본 데이터 분석, 전처리 및 AI 모델을 이용하여 CNC 공작기계에서 발생하는 불량 가공품의 실시간 감지 및 피쳐 중요도에 따른 신속한 피쳐별 점검을 통해 지속적인 불량품 양산을 사전에 예방함으로써 자동차 부품 등 제조 기업의 비용 절감 및 영업이익 증대에 기여함</li> </ul> </li> </ul>
<p><b>현장지원 소감</b></p>	<ul style="list-style-type: none"> <li>○ 파견을 계기로 인공지능이라는 매개체를 통해 전자공학 분야를 넘어 기계공학 분야와 전자공학 분야의 융합이 일어나는 기업 현장의 니즈를 경험하고 산업현장의 문제 해결을 위해서는 전공분야 간 융합이 필요하다는 점을 깨닫는 유익한 계기가 되었음</li> <li>○ 파견 기업이 일반적인 제조 분야의 중소기업이 아닌 제조 솔루션 제공 기업인 관계로 AI를 적용하기에는 현장 데이터의 부재라는 기본적인 문제가 있고 데이터를 보유한 고객기업 현장의 입장에서는 데이터 자체의 공유가 기업 보안상 극히 어렵기 때문에, 제조 AI의 경우에 데이터 확보라는 기본 전제의 충족부터가 어려운 점을 절감함</li> </ul>