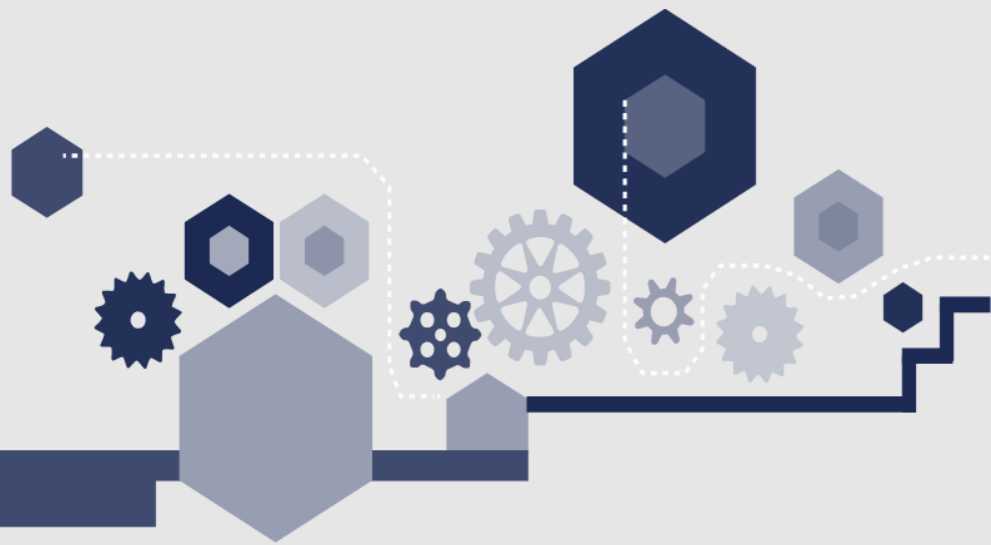


# preLab 5차

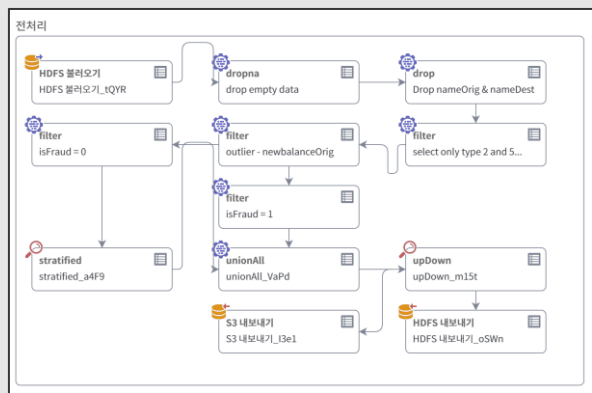


# 순서

- ✓ 1. 개요
- ✓ 2. Modeling Review
- ✓ 3. 자산화
  - 배포 관리
  - 소스 관리
  - 분석이미지 관리
- ✓ 4. Automated Machine Learning
- ✓ 5. pre Lab 총정리

# 개요

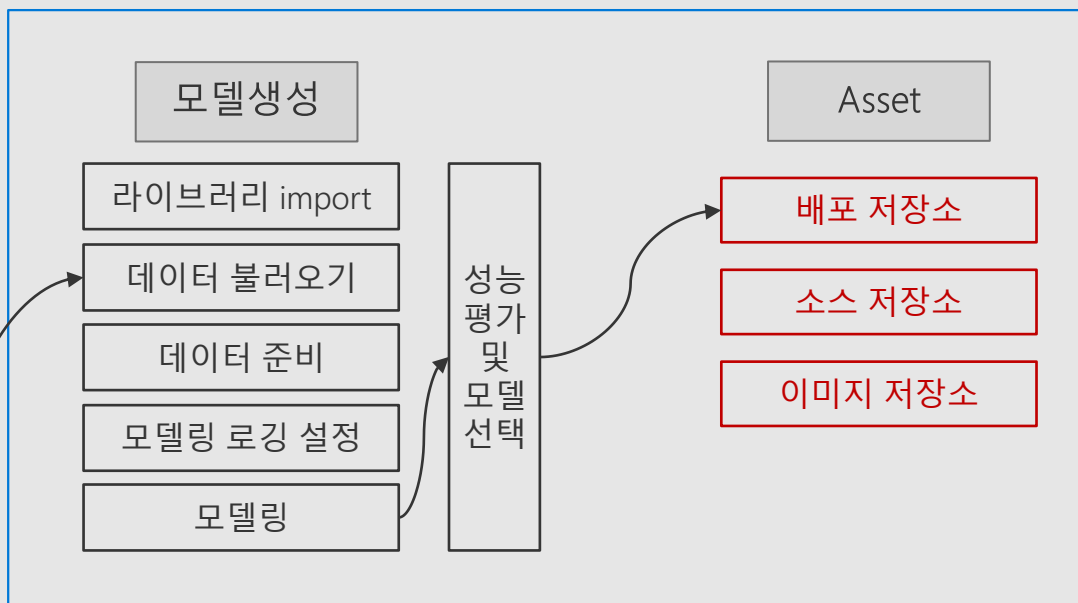
## Pipeline



HDFS  
S3

csv file

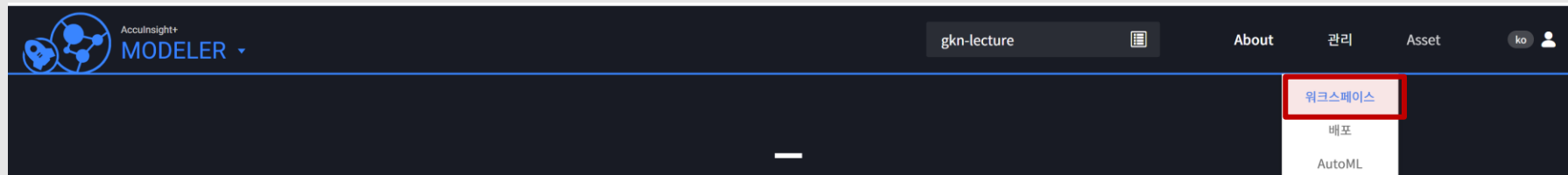
## Modeler



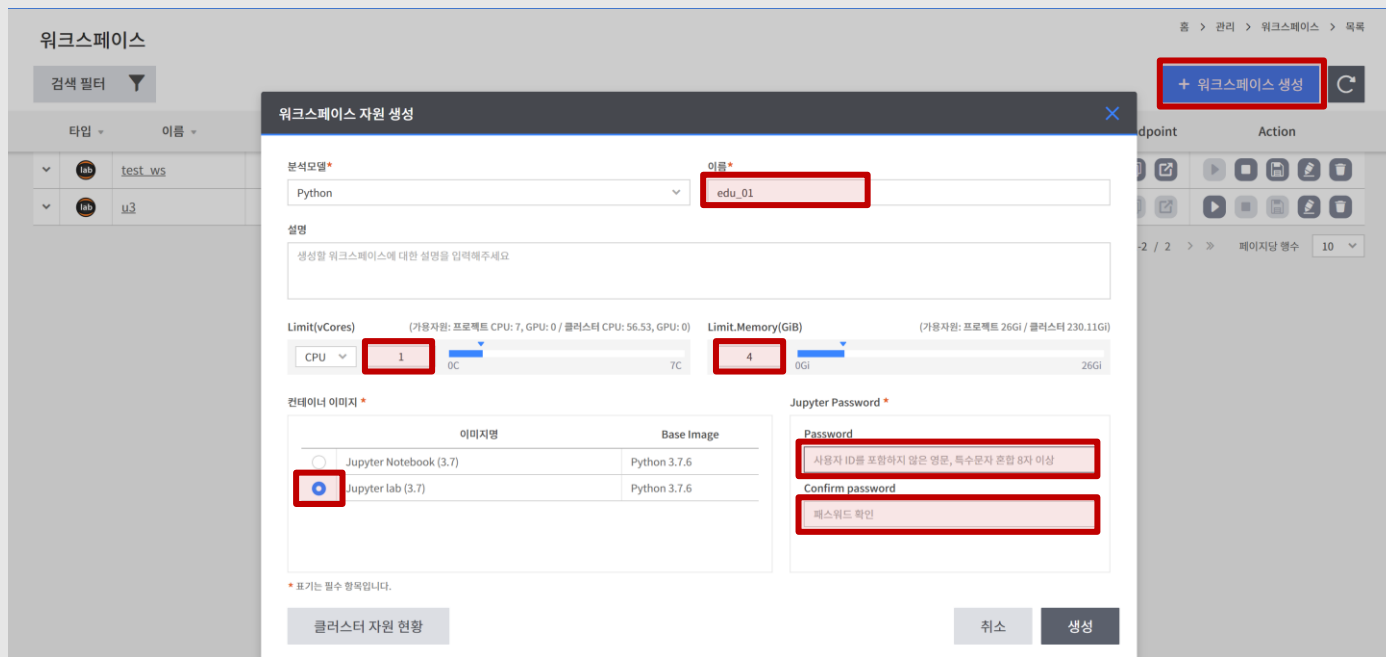
# Modeling Review

# 개요 - 준비사항

✓ Modeler 로그인



✓ 워크스페이스 생성



# Modeling Using Accuinsight Modeler

## ✓ 코드 구조

필요한 라이브러리 import

데이터 불러오기

데이터 준비

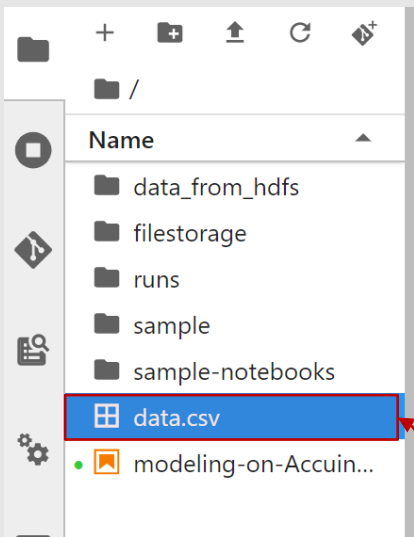
모델링 로깅 설정

모델링

# 데이터 가져오기

## ✓ S3로 부터 데이터 가져오기

- boto 패키지 설치
- aws\_access\_key\_id, aws\_secret\_access\_key
- S3에 저장된 파일 경로, 저장할 파일 경로



### 2) from S3

- aws\_access\_key\_id, aws\_secret\_access\_key 확인 필요

```
# s3에 접속하여 데이터 다운로드하기 위한 패키지  
!pip3 install boto3
```

필요한 패키지 설치

```
import boto3
```

```
# Hard coded strings as credentials, not recommended.  
accessKey='AKIA2HASEMBXXCLOKFMP'  
secretKey='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'
```

Access key, 경로  
저장할 파일이름 지정

```
client = boto3.client('s3', aws_access_key_id=accessKey, aws_secret_access_key=secretKey)
```

```
bucket_name = 'accu-trial' # s3 버킷 명
```

```
file_name = 'data.csv' # 저장할 경로 및 파일명
```

```
key = 'accu-trial03/LabTest/pre_processed_data/part-00000-1b58514c-e0ae-43c9-b9d1-1a8a0049a0cb-c000.csv' # 다운로드할 파일경로
```

```
session = boto3.Session(aws_access_key_id=accessKey,aws_secret_access_key=secretKey)
```

```
s3 = session.resource('s3')
```

```
client.download_file(bucket_name, key, file_name)
```

Download

# 실습 #1

## ✓ 워크스페이스 생성

- 워크스페이스가 없는 사람은 새로 생성하십시오.

## ✓ 제공된 소스코드를 단계별로 수행하며 Review를 합니다.

- modeling-on-Accuinsight.ipynb
- 데이터 가져오기 :
  - S3로 부터 파일을 다운받아 사용합니다.
- 모델링 :
  - 모델의 하이퍼파라미터를 조정하여 적어도 2개 이상을 생성합니다.
  - 모델링( .fit 구문)을 수행할 때마다, 반드시 `accu.autolog()` 구문을 먼저 실행해야 합니다.
- 성능 평가 : 성능을 비교해보고 제일 좋은 모델을 선정합니다.



# Accuinsight Modeler와 개인 개발도구 활용

- ✓ 활용 편리성 + 비용 고려
- ✓ 개인 개발도구 : Anaconda, Google Colab
  - 데이터 정제, EDA&CDA
  - 초기 모델 개발 및 다양한 테스트 용
- ✓ Accuinsight
  - Pipeline 구성
  - 정리된 모델 개발 및 성능 테스트
  - 배포, 소스관리 등

자산화

# 배포 관리

✓ 성능 비교 후 선정된 모델을 선택 > 배포 저장소 이동

AccuInsight+ MODELER

gkn-lecture About 관리 Asset ko

< test\_ws 상태 RUNNING

요약 Experiment 작업폴더 계정폴더

run name을 입력하세요

Comparison Edit Columns 배포 저장소 이동

	№	Run ID	Executor	Update On	Version	
					data	model
<input type="checkbox"/>	1	tf.keras-07CE7DEC9E06483D...	trial_user03	2021-04-26 14:05:49		8fef655b00a0a1de6df0484705a77812f4c0465c
<input type="checkbox"/>	2	tf.keras-181E29E79E6B444A...	trial_user03	2021-04-26 14:09:45		c1f8dc25131ce5288b5c93865f6ff2cd4117938f
<input type="checkbox"/>	3	tf.keras-20CD71298A0A4B6E...	trial_user03	2021-04-26 18:27:18	hdfs://10.31.200.106/8020/user/hadoop/LabTes...	f5dc3b08bfc3f47a7e8ad84c025e99f16ad2da6b
<input type="checkbox"/>	4	tf.keras-414A4CEA081E4293...	trial_user03	2021-04-26 19:24:06	hdfs://10.31.200.106/8020/user/hadoop/LabTes...	a4b8c89c17954cb54a2515ef71a17e94e2605fa9
<input type="checkbox"/>	5	tf.keras-D524B38153CB4465...	trial_user03	2021-04-26 19:28:20	hdfs://10.31.200.106/8020/user/hadoop/LabTes...	8615a827210af9c692b86e867cf4f0ae9b32a9a3
<input checked="" type="checkbox"/>	6	tf.keras-DAF012272F0241BC...	trial_user03	2021-04-27 16:07:09	hdfs://10.31.200.106/8020/user/hadoop/LabTes...	0b8d89e4e4954dff6ebdc62086e2738ae51b8b54

AccuInsight+ MODELER

gkn-lecture About 관리 Asset ko

배포 저장소

검색 필터

홈 > 소스저장소 > 목록

배포저장소

분석이미지

	타입	이름	워크스페이스명	마지막 업데이트 메시지	생성일	마지막 수정	Owner	API	배포	Action
<input type="checkbox"/>		tf.keras-DAF012272F0241BC...	test_ws		2021.04.28 08:30:50	2021.04.28 08:31:13	trial_user03			

# 배포 관리

< tf.keras-92A18DB39CA64A4FA596268...



요약

히스토리

프로젝트	gkn-lecture
모델명	tf.keras-92A18DB39CA64A4FA596268A8E7F3FC5_9_model
최종 버전	-
모델 설명	-
인자 설명	-
최종 작성자	-
생성 시간	2021.04.24 19:31:05
워크스페이스 정보	accu-edu01_ws [설치 패키지] cat:_nullnull:

파일

- ▶ runs
- .packages
- fashion-mnist-experiment.ipynb

배포 모델 버전 선택

버전 선택

버전 ▼	코멘트	업데이트 ▼	업데이트 계정
<input checked="" type="radio"/> v1.0.0	tf.keras-414A4CEA081E4293AD01895	2021-04-27 05:01:08	trial_user03@s...

취소

확인

# 배포 관리

## ① 배포 정보

### 배포 정보

배포명

test-deploy

AutoScaling 최대 개수

1

Limit(vCores) (가용자원: 프로젝트 CPU: 6, GPU: 0 / 클러스터 CPU: 82.35, GPU: 0)

CPU ▾

0.5

0C

6C

Limit.Memory(GiB)

(가용자원: 프로젝트 23Gi / 클러스터 326.67Gi)

1

0Gi

23Gi

## ② RESTAPI 코드

### RESTAPI.PY

```
1 #####
2 # Do not modify this code block #
3 # Import monitoring packages
4 import logging, traceback
5 from Accuinsight.Monitoring.deploy.monitoring_deploy import AddDeployLog
6
7 # Import flask packages
8 from flask import Flask, make_response, request
9 from flask_restplus import Api, Resource
10 from flask_cors import CORS
11 #####
12
13 # Custom packages
```

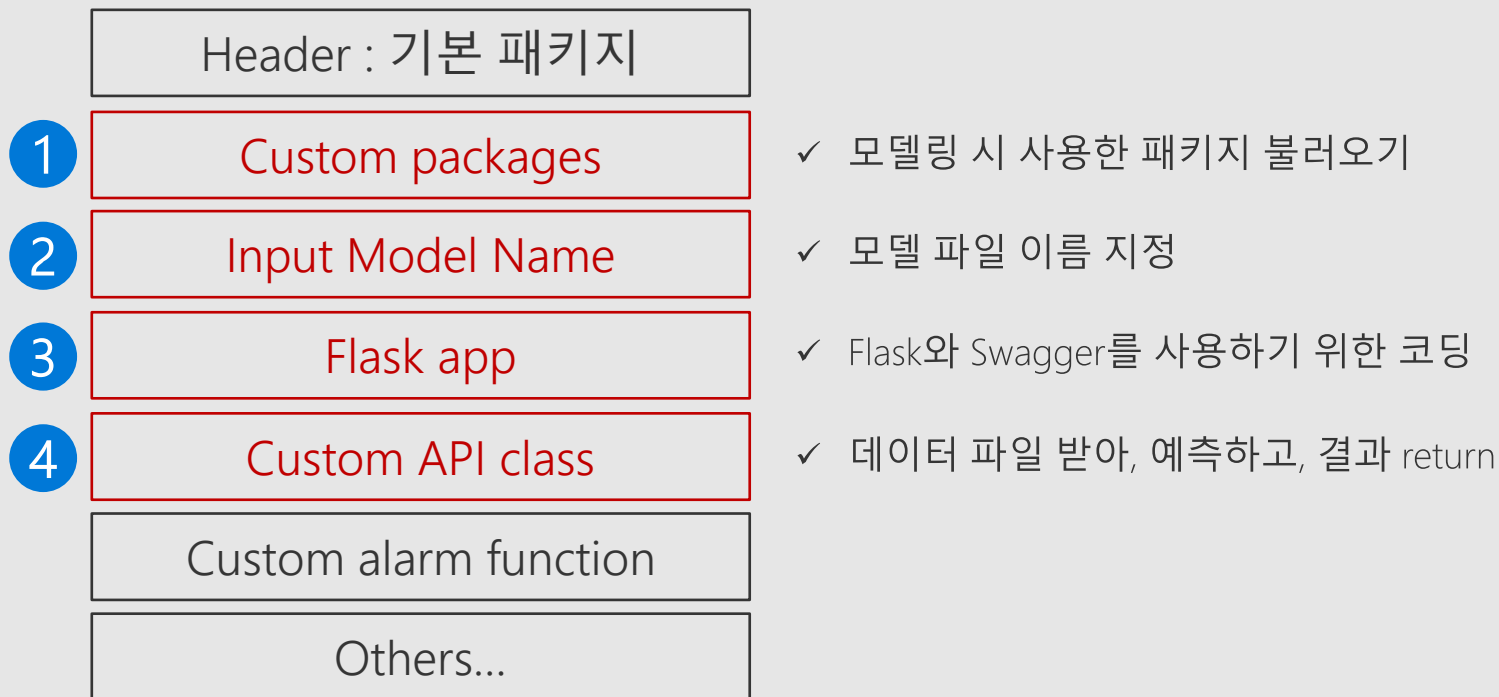
# 배포 관리

## ✓ RESTAPI 코드

- RESTAPI 코드를 수정해야 합니다.
- 우선 코드 구조를 설명하고
- 수정할 부분을 짚겠습니다.
- Jupyter Lab에서 수정하는 것이 편리합니다.

# 배포 관리

## ✓ RESTAPI 코드 구조



# 배포 관리 : ①Custom packages

✓ 모델링에 사용한 패키지들을 추가합니다.

- Rest API에서 필요

```
# Custom packages -----  
from werkzeug.datastructures import FileStorage  
import numpy as np  
import pandas as pd  
from tensorflow.keras.models import model_from_json
```

- 모델링에서 필요.

```
import tensorflow as tf  
from tensorflow import keras  
from tensorflow.keras import layers  
from datetime import date  
import json  
import os  
from collections import OrderedDict  
  
from Accuinsight.Lifecycle.tensorflow import accuinsight
```


```
# -----
```




# 배포 관리 : ②Input Model Name



```
# Input Model Name -----  
model_name = 'tf.keras-DAF01227-2F02-41BC-8DA2-91ABEDC9D1F6-epoch-00010-val_accuracy-0.9895312786102295'  
# -----
```

배포 저장소

검색 필터 

홈 > Asset > 배포저장소 > 목록

삭제 

<input type="checkbox"/>	타입	이름	워크스페이스명	마지막 업데이트 메시지	생성일	마지막 수정	Owner	API	배포	Action
<input type="checkbox"/>		tf.keras-6A33DD70DF4B426...	test_ws							
<input type="checkbox"/>		tf.keras-DAF012272F0241B...	test_ws							

요약

히스토리

최종 작성자

-

생성 시간

2021.04.28 08:30:50

워크스페이스 정보

test\_ws

[설치 패키지]

cat:\_nullnull:

파일

runs

best-model

tf.keras-DAF01227-2F02-41BC-8DA2-91ABEDC9D1F6-epoch-00010-val\_accuracy-0.9895312786102295.h5

tf.keras-DAF01227-2F02-41BC-8DA2-91ABEDC9D1F6-epoch-00010-val\_accuracy-0.9895312786102295.json

.packages

• 확장자 제외한 파일 이름을 붙여 넣는다.

# 배포 관리 : ③ Flask app

## ✓ 수정 가능한 영역

```
# Flask app
app = Flask(__name__)
api = Api(app, version='1.0', title='REST API', doc='/__swagger__', description='A sample API for demonstration')
app.config.SWAGGER_UI_DOC_EXPANSION = 'list'
ns_conf = api.namespace('fashion-deploy', description='AI Model Deploy Example')
upload_parser = ns_conf.parser()
upload_parser.add_argument('test_data', location='files',
                           type=FileStorage, required=True)
upload_parser.add_argument('test_label', location='files',
                           type=FileStorage, required=True)
```

test-deploy 상태 RUNNING

요약 예측이력 소스 테스트 로그(API)

**REST API** 1.0  
[ Base URL: /modeler/gateway/rpmodeler/17 ]  
./swaggerjson  
A sample API for demonstration

**fashion-deploy** AI Model Deploy Example

**POST** /fashion-deploy/

Parameters

Name

**test\_data** \* required  
file  
{ formData }

**test\_label** \* required  
file  
{ formData }

# 배포 관리 : ④ Custom API class

```
# Custom API class -----  
@ns_conf.route("/")  
class ConferenceList(Resource):  
    @staticmethod  
    @ns_conf.expect(upload_parser)  
    def post():  
        # Load model from json  
        print(os.getcwd())  
        dir_fd = os.open('runs/best-model/', os.O_RDONLY)
```

모델 로딩

```
        def opener(path, flags):  
            return os.open(path, flags, dir_fd=dir_fd)  
  
        json_file = open(model_name+'.json', 'r', opener=opener)  
  
        loaded_model_json = json_file.read()  
        json_file.close()  
        loaded_model = model_from_json(loaded_model_json)
```

Test data 로딩

```
        #load weights into new model  
        loaded_model.load_weights('runs/best-model/' + model_name + '.h5')  
        if upload_parser.parse_args():  
            data = upload_parser.parse_args().pop('test_data')  
            label = upload_parser.parse_args().pop('test_label')  
            test_data = pd.read_csv(data)  
            label = pd.read_csv(label)
```

예측하기

```
            test_data = np.array(test_data.iloc[0])  
            test_y = label.iloc[0].tolist()  
            test_pred = loaded_model.predict(test_data).tolist()[0].pop()
```

결과 return

```
            return 'target: ' + str(test_y) + " predicted value: " + str(test_pred)  
        else:  
            raise FileNotFoundError
```

# 배포 관리 : 테스트

배포

홈 > 관리 > 배포 > 목록

검색 필터

타입	이름	Endpoint	모델명	레플리카셋	배포 일시	Owner	상태	Action
	test-dep	https://accuinsight.cloudz.co.kr/modeler/...	tf.keras-FAA572C06F6B4...	1 / 1	2021-04-29 11:28:56	trial_user03	RUNNING	

Accuinsight+  
MODELER

gkn-lecture

About관리Asset

< ttt111 상태 RUNNING

홈 > 관리 > 배포 > 상세

요약예측이력소스테스트로그(API)

Fraud-Detection-deploy AI Model Deploy Example

POST /Fraud-Detection-deploy/

Parameters

NameDescription

test\_data \* required  
file  
(formData)

파일 선택선택된 파일 없음

test\_label \* required  
file  
(formData)

파일 선택선택된 파일 없음

Execute

# 배포 관리 : 테스트

## Responses

Response content type

application/json



### Curl

```
curl -X POST "https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/30/Fraud-Detection-deploy/" -H "accept: application/json" -H "Content-Type: multipart/form-data" -F "test_data=@test_x.CSV:type=application/vnd.ms-excel" -F "test_label=@test_y.CSV:type=application/vnd.ms-excel"
```

### Request URL

https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/30/Fraud-Detection-deploy/

### Server response

#### Code

#### Details

200

#### Response body

"target: 0, predicted value : 0"

Download

#### Response headers

```
content-encoding: gzip
content-type: application/json
date: Wed, 28 Apr 2021 09:30:50 GMT
server: nginx
vary: Accept-Encoding, Accept-Encoding
```

# 배포관리 - 테스트

✓ 테스트 탭이 아니라, 인터넷 브라우저로 실행 가능

< test-dep 상태 RUNNING ▶ □ 🔍 🗑️ 🔄

요약

예측이력

소스

테스트

로그(API)

배포명	test-dep
프로젝트	gkn-lecture
생성 시간	2021-04-29 11:28:56
배포 상태	<span>RUNNING</span>
AutoScaling 최대 개수	1
Resource	Limit.CPU: 1.00C, Limit.GPU: 0.00C, Limit.Memory: 1.00Gi
모델명	tf.keras-FAA572C06F6B4DD09154C163BBBEF8A5_89_model
Version	
Endpoint	<a href="https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/">https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/</a>
인증키	76125954-6553-4c44-919f-6495d3c57228 <span>임시 인증키 생성</span>
테스트 Endpoint	<a href="https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/_swagger_...">https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/_swagger_...</a> <span>주소 복사</span>

레프리카세 (총 1 Running)

REST API

[https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/\\_swagger\\_...](https://accuinsight.cloudz.co.kr/modeler/gateway/rpmodeler/31/_swagger_...)

REST API 1.0

[ Base URL: /modeler/gateway/rpmodeler/31 ]  
[./swagger.json](#)

A sample API for demonstration

Fraud-Detection-deploy AI Model Deploy Example

POST /Fraud-Detection-deploy/

Parameters

Name	Description
test_data * required	
file (formData)	<input type="text" value="파일 선택 test_x.CSV"/>
test_label * required	
file (formData)	<input type="text" value="파일 선택 test_y.CSV"/>

# 배포 관리 : 예측이력 조회

홈 > 관리 > 배포 > 상세

< test-dep

상태 RUNNING



요약

예측이력

소스

테스트

로그(API)

URL

API 검색

요청 Method

Request Method 검색

응답코드

결과 코드 검색



API ▾

시작시간 ▾

종료시간 ▾

실행시간

성공여부 ▾

Method ▾

Response

/Fraud-Detection-deploy/

2021-04-29 11:37:18

2021-04-29 11:37:18

0.457

Success

POST

target: 0, predicted value : 0



« < 1-1 / 1 > »

페이지당 행수

10 ▾



Success: 100.00%

Total 1

Success 1

Fail 0

## 실습#2 : 배포

- ① 워크스페이스의 'Experiment' 탭에서 모델을 선정한 후 '배포 저장소로 이동' 시킵니다.
- ② 배포저장소에서, 항목을 선택하고 세부정보로 들어갑니다.
- ③ 배포 정보 입력
- ④ REST API 코드 수정
- ⑤ 배포 실행 (약 5~10분 정도 시간이 소요됩니다.)
- ⑥ 테스트 페이지에서 테스트 수행
  - 제공 파일 중 test\_x.csv 와 test\_y.csv 파일을 이용하십시오.
- ⑦ 예측 이력을 확인해 봅시다.



# 소스 저장소

✓ 모델러 > 관리 > 워크스페이스 > 상세

- 주의사항: 파일이름 규칙 - 한글, 스페이스, \_ 등 안됨.

The screenshot shows the Databricks workspace interface. At the top, there's a breadcrumb navigation: < test\_ws > 상태 RUNNING > 실행 > 중지 > 새로고침 > 다시 실행. Below this, there are tabs: 요약, Experiment, **작업폴더** (highlighted with a red box), and 계정폴더. The main area shows the file system under /home/work (Notebook Working 디렉토리 파일 목록). A list of files and folders is shown, including .ipynb\_checkpoints, .virtual\_documents, data\_from\_hdfs (checked), filestorage, runs, sample, sample-notebooks, .env, 2.샘플링과 모델링\_연습.ipynb, 3.모델링 on Accuinsight.ipynb, and Class-Imbalance.ipynb (checked). A modal titled '소스 저장소 저장' (Save Source Storage) is open, showing the file path 'data\_from\_hdfs/part-00000-d73dd82a-42ae-4608-ac6a-4b36b671909b-c000\_20210427.csv외3', the option to '신규 저장소 추가' (Add new storage), the source name 'test-source', and the description 'test'. A red arrow points from the '소스 저장소 저장' button in the top right to the modal. The modal has '취소' (Cancel) and '저장' (Save) buttons.

홈 > 관리 > 워크스페이스 > 상세

요약 Experiment **작업폴더** 계정폴더

**/home/work** (Notebook Working 디렉토리 파일 목록)

소스 저장소 저장

배포 저장소로 저장

data\_from\_hdfs/part-00000-d73dd82a-42ae-4608-ac6a-4b36b671909b-c000\_20210427.csv외3

☒ 신규 저장소 추가 ☐ 저장된 소스 업데이트

소스명

test-source

설명

test

취소 저장

파일보기

파일보기

파일보기

파일보기

# 소스 저장소

✓ 모델러 > Asset > 소스저장소 > 목록




## 소스 저장소

홈 > Asset > 소스저장소 > 목록

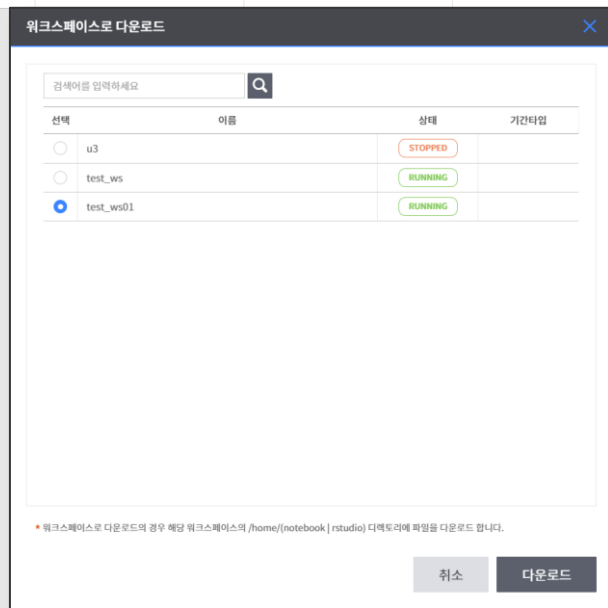
검색 필터

삭제



<input type="checkbox"/>	이름 ▾	워크스페이스명	마지막 업데이트 메시지	생성일 ▾	마지막 수정 ▾	Owner	Action
<input type="checkbox"/>	<a href="#">test-source</a>	test_ws	test	2021.04.28 11:58:40	2021.04.28 11:59:23	trial_user03	  

- 다른 워크스페이스에 소스코드를 복사하기



# 실습#3 : 소스 관리

- ① 워크스페이스 > 작업폴더 탭에서 저장할 소스를 선택합니다. 소스와 데이터파일 등을 선택
- ② 소스 저장소로 저장 > 신규저장소 저장 (시간이 좀 걸립니다.)
- ③ 새 워크스페이스를 생성합니다.
- ④ 소스 저장소로 와서 새 워크스페이스로 소스를 저장시킵니다.
- ⑤ 새 워크스페이스 Jupyter Lab에 접속하여 복사된 소스를 확인합니다.

# 분석 이미지 관리 - 분석환경 저장하기

✓ 모델러 > 관리 > 워크스페이스

✓ 저장을 하면, 분석환경 전체를 이미지 저장소에 저장.

워크스페이스 컨테이너 이미지 저장

이름  
preLab4

이미지 (이미지명은 영문 소문자, 숫자 또는 특수문자만 사용 가능합니다)  
preLab4 prelab4

설명  
preLab4에서 사용한 소스코드, 데이터

공유  
프로젝트 공유

워크스페이스명  
test\_ws

Base Image  
/prd/modeller/workspace/py3.7-cuda10.1:workspace\_v2.8.19

\* 계정당 컨테이너 이미지는 최대 n개까지 생성할 수 있습니다

취소 추가

워크스페이스

검색 필터

타입 이름

4-26 0... trial\_user03 2 1일

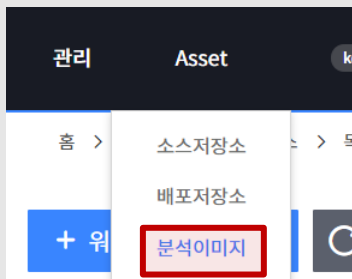
Owner 재시작 기간타임 Endpoint Action

+ 워크스페이스 생성

컨테이너 이미지 저장

- 프로젝트 공유 : 프로젝트 멤버들에게만 공유
- 서비스 전체 공유 : 모든 모델러 사용자에게

# 분석 이미지 관리



### 워크스페이스 이미지

이미지 관리

공유 이미지

이름 ▾	설명	이미지명	상태 ▾	공유	Type	생성일 ▾	Action
preLab4	preLab4에서 사용한 소스코드, 데이터	preLab4:prelab4	UPLOADING	PROJECT	Jupyter lab (3.7)	2021-04-28 10:5...	  

이미지 관리

공유 이미지



프로젝트 ▾	이름 ▾	설명	이미지명	상태 ▾	공유	Type	생성일 ▾	Action
gkn-lecture	preLab4	preLab4에서 사용한 소스코드...	preLab4:prelab4	UPLOADING	PROJECT	Jupyter lab (3.7)	2021-04-28 10:5...	

# 분석 이미지 관리 - 분석이미지 활용

워크스페이스

검색 필터

홈 > 관리 > 워크스페이스 > 목록

+ 워크스페이스 생성

타입

이름

설명

워크스페이스 자원 생성

Action

분석모델\*

Python

이름\*

생성할 워크스페이스 이름을 입력해주세요.

설명

생성할 워크스페이스에 대한 설명을 입력해주세요.

Limit(vCores)

(가용자원: 프로젝트 CPU: 6, GPU: 0 / 클러스터 CPU: 21.13, GPU: 0)

CPU

0

0C

6C

Limit.Memory(GiB)

(가용자원: 프로젝트 27Gi / 클러스터 154.27Gi)

0

0Gi

27Gi

컨테이너 이미지 \*

이미지명		Base Image
<input type="radio"/>	Jupyter Notebook (3.7)	Python 3.7.6
<input type="radio"/>	Jupyter lab (3.7)	Python 3.7.6
<input type="radio"/>	test_sw	Jupyter lab (3.7)
<input type="radio"/>	ch-img01	Jupyter lab (3.7)

Jupyter Password \*

Password

사용자 ID를 포함하지 않은 영문, 특수문자 혼합 8자 이상

Confirm password

패스워드 확인

\* 표기는 필수 항목입니다.

클러스터 자원 현황

취소

생성

신규 워크스페이스 생성시  
컨테이너 이미지에서 사용가능

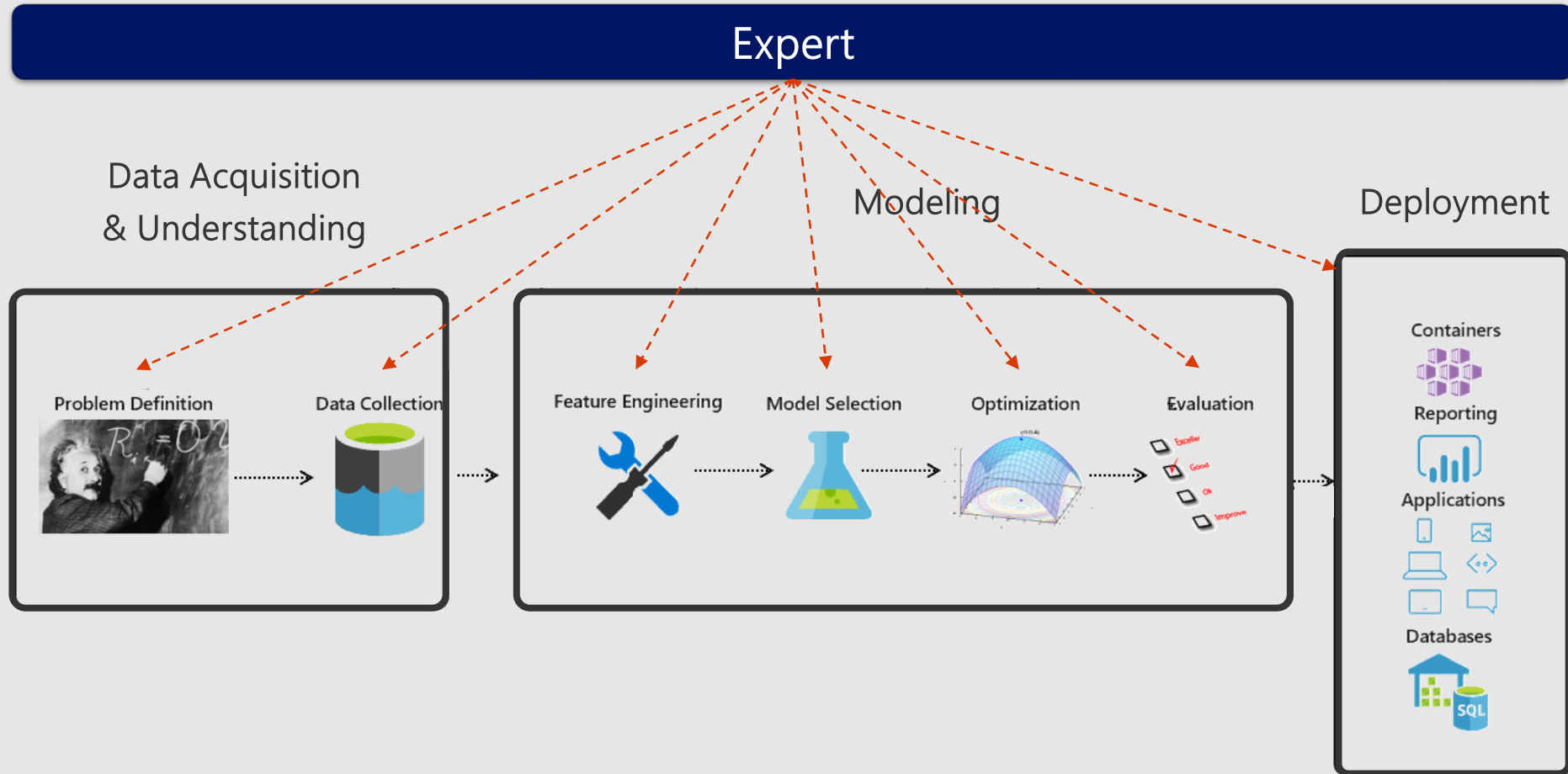
# 실습#4 : 분석 이미지 관리

- ① 워크스페이스 목록에서, 하나를 정해 저장
- ② 관리 > 분석 이미지, 로 이동
- ③ 저장된 분석 이미지 확인
- ④ 신규 워크스페이스 생성 : 컨테이너 이미지에서 저장된 분석이미지 선택

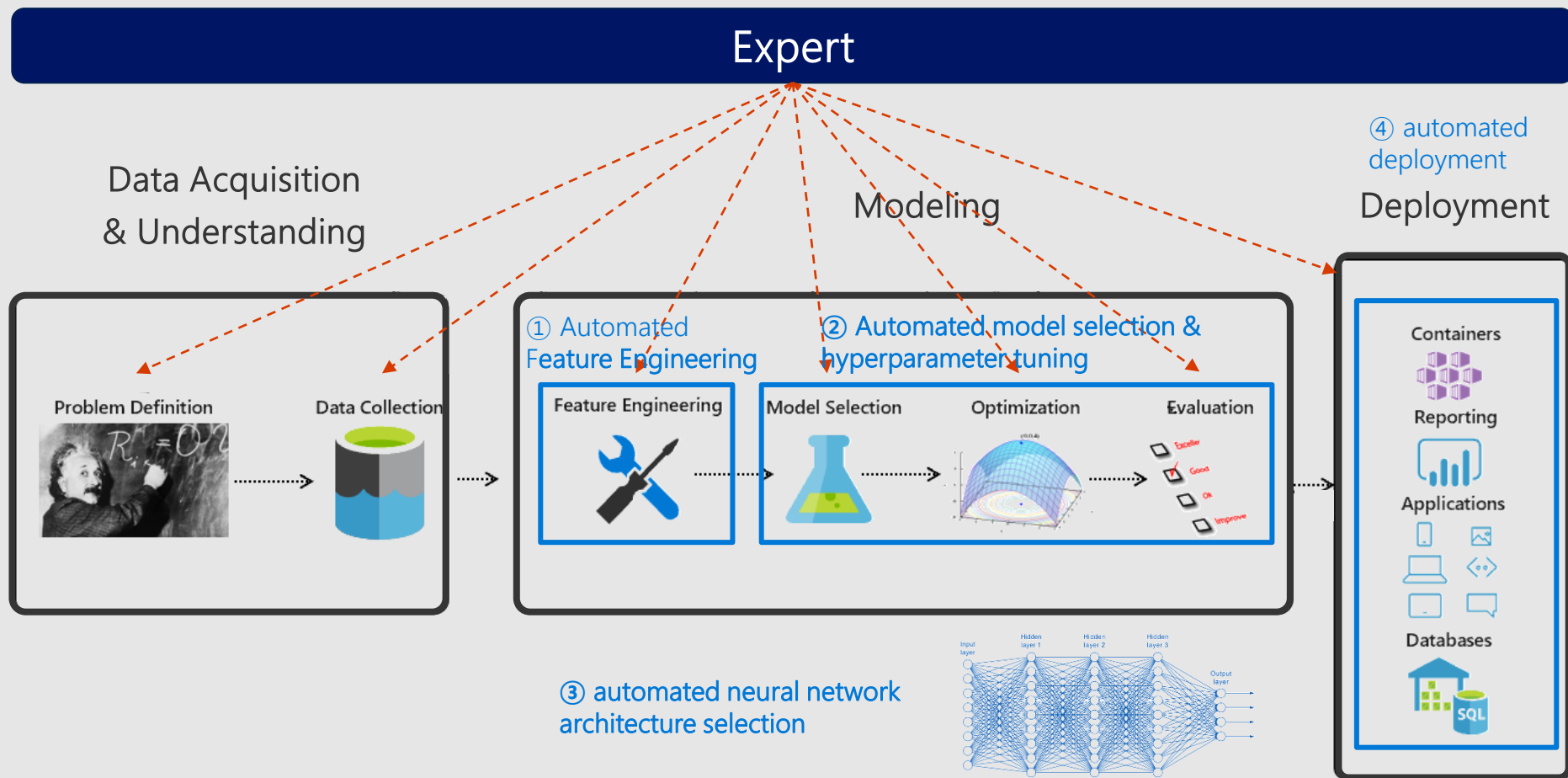
# Automated Machine Learning



# Machine Learning Pipeline (ML Process)



# Auto ML



# AutoML

## ✓ AutoML의 일반적인 범위

- ① automated feature engineering
- ② automated model selection and hyperparameter tuning
- ③ automated neural network architecture selection
- ④ automated deployment