

2-17. 지능형 곤충 사육 데이터

AI 모델환경 설치 가이드

v1.0

사용자 매뉴얼



목 차

Chapter 1. 프로그램 환경 구성

1-1. 프로그램 개요	3
1-2. 프로그램 요구사항	3
1-3. 프로그램 환경 구성.....	3

Chapter 2. 프로그램 기능 설명

2-1. 프로그램 사용방법	6
----------------------	---

1. 프로그램 환경 구성

1-1 프로그램 개요

이 프로그램은 인공지능 학습용 데이터 구축 사업(2-17. 지능형 곤충 사육 데이터)에서 구축한 인공지능 학습용 데이터를 활용해 곤충의 성장 수치를 예측하는 AI 모델 생성 및 모델 활용하는 프로그램이다.

인공지능 학습용 데이터는 총 4 종의 곤충(누에, 갈색거저리, 흰점박이꽃무지, 동애등애)의 4 가지 생육단계(알, 애벌레, 번데기, 성충)로 구성되어 있다. AI 모델은 4 종 곤충의 4 개 생육단계, 라벨링데이터(곤충길이, 무게), 모델(랜덤포레스트, 선형 SVM)별로 총 64 개 모델로 만들어져 있다.

인공지능 학습용 데이터

인공지능 학습용 데이터는 다음과 같은 형식의 json 파일이다. 곤충의 성장 수치를 예측하는 AI 모델 개발에는 json 파일의 데이터 중에서 ANNOTATION(영상 부분)은 제외한 메타 데이터(META), 센서 데이터(SENSOR), 관리데이터(MANAGEMENT)를 사용한다.

```
{
  "META": { ...
},
  "ANNOTATION": { ...
},
  "SENSOR": { ...
},
  "MANAGEMENT": { ...
}
```

< 지능형 곤충 사육 데이터의 형식>

학습용 데이터의 독립변수

변수명	위치	비고
Breeding Room Indoor Temperature	SENSOR	실내온도
Breeding Room Indoor Humidity	SENSOR	실내 습도

Breeding Room Indoor Co2	SENSOR	실내 CO2 농도
Breeding Room Indoor Surface Moisture Quantity	SENSOR	실내 습도
Breeding Room Indoor Illumination Intensity	SENSOR	실내 조도
Breeding Room Indoor Ammonia	SENSOR	실내 암모니아 농도
Breeding Room Indoor Oxygen	SENSOR	실내 산소 농도
Breeding Room External Air Velocity	SENSOR	실외 풍속
Breeding Room External Temperature	SENSOR	실외온도
Breeding Room External Humidity	SENSOR	실외 습도
Insect Breeding Box Temperature	SENSOR	사육상자 온도
Insect Breeding Box Humidity	SENSOR	사육상자 습도
Insect Breeding Box Co2	SENSOR	사육상자 CO2 농도
Insect Breeding Box Mat Temperature	SENSOR	사육상자 매트 온도
Insect Length	MANAGEMENT	곤충 길이
Insect Average Weight	MANAGEMENT	곤충 평균 무게

학습용 데이터의 종속변수(라벨링 데이터)

변수명	위치	비고
Insect Length	MANAGEMENT	최종 곤충 길이
Insect Average Weight	MANAGEMENT	최종 곤충 평균 무게

성장 예측 모델

성장 예측 모델 64 개는 파이썬 Pandas, sklearn 을 사용해 개발하였으며 사용한 모델은 Random Forest, leaner SVM 이다.

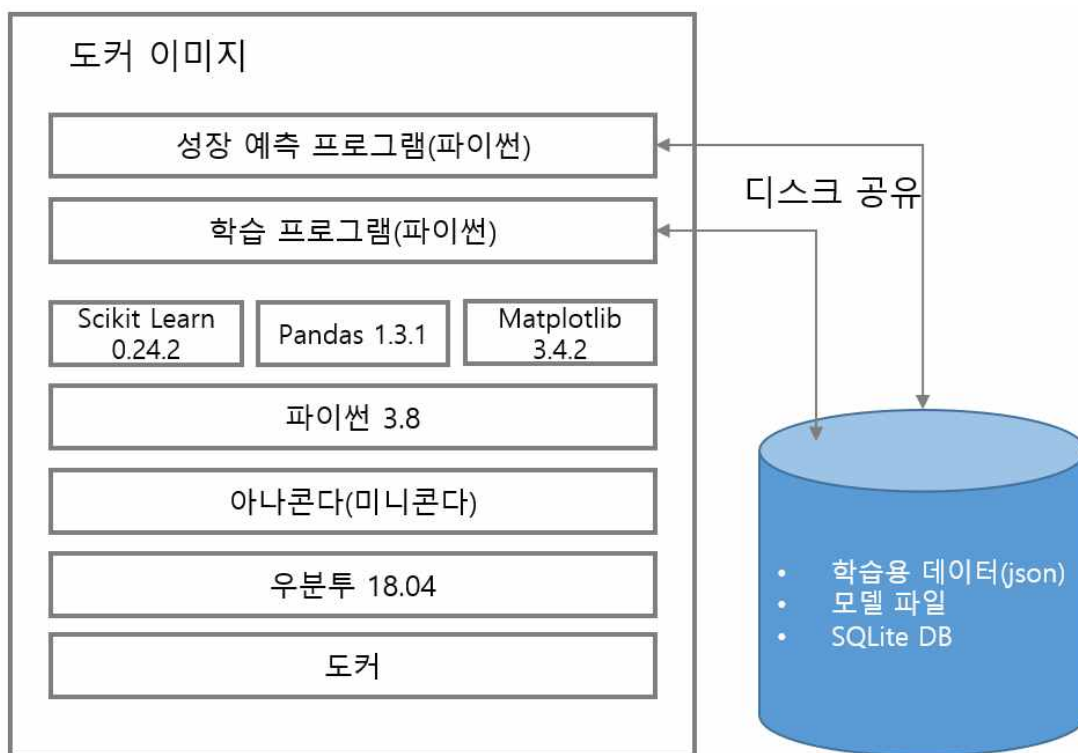
1-2 프로그램 요구사항

곤충의 성장 수치를 예측 프로그램은 파이썬으로 만들어졌으며 도커(docker) 이미지로 제공되기 때문에 이 프로그램을 설치하려면 사용자 PC에 도커가 설치되어 있어야 하며 프로그램 테스트를 위해서는 파이썬 3.X가 필요하다.

	요구사항	비고
프로그램 설치	도커	도커 이미지(파이썬 프로그램 포함)
프로그램 테스트	파이썬 3.X	테스트용 파이썬 프로그램

1-3 프로그램 환경구성

성장 예측 프로그램은 다음과 같이 구성된다. 도커 이미지에는 이미 아나콘다, 파이썬 및 Scikit learn, Pandas, Matplotlib 등의 패키지가 설치되어 있기 때문에 별도의 패키지 설치의 필요없다. 도커 이미지를 실행할 수 있는 도커 프로그램만 필요하다.



< 환경 구성 >

학습을 통해 만든 모델을 저장하기 위해 호스트의 특정 디렉토리를 공유한다. (도커 이미지는 읽기 전용이기 때문에 데이터 저장이 불가능하다.) 그리고 이렇게 만든 공유 디렉토리의 모델은 성장 예측 프로그램에서 사용한다.

2. 프로그램 기능 설명

2-1 프로그램 사용방법

호스트 준비 사항

도커이미지를 이용한 모델 생성, 검증 작업등의 결과는 도커 이미지에 저장이 불가능하다. 따라서 도커를 실행중인 호스트에 결과를 저장하기 위해 호스트 볼륨을 미리 준비한다. 검증 도커 컨테이너에서 사용할 호스트 볼륨(/usr/local/insect/src)에 미리 필요한 파일들을 저장해 둔다.

```
total 36
drwxr-xr-x 7 root root 4096 Jan 18 02:25 .
drwxr-xr-x 6 root root 4096 Jan 18 01:14 ..
drwxr-xr-x 3 root root 4096 Jan 18 02:27 품질검증
-rw-r--r-- 1 root root 2560 Jan 18 02:18 '사육시작끝.csv'
-rw-r--r-- 1 root root 71 Jan 18 02:03 '사육시작끝.json'
drwxr-xr-x 2 root root 4096 Jan 18 01:29 log
drwxr-xr-x 2 root root 4096 Jan 18 02:48 model
drwxr-xr-x 2 root root 4096 Jan 18 02:48 sql
drwxr-xr-x 2 root root 4096 Jan 18 01:27 src
root@ubuntu:/usr/local/src/study/insect# tree model/
```

<도커 호스트 볼륨>

디렉토리명	내용	비고
품질검증	json 파일들이 저장된 디렉토리	반드시 json 파일들을 미리 저장해둔다.
log	로그 파일 디렉토리	빈 디렉토리로 만들어둔다.
model	모델을 저장할 디렉토리	빈 디렉토리로 만들어둔다.
sql	sqlite db 를 저장할 디렉토리	빈 디렉토리로 만들어둔다.
src	소스코드 디렉토리	빈 디렉토리로 만들어둔다. 실제 소스를 도커이미지의 소스를 사용하며 이 디렉토리는 나머지 디렉토리의 상대 경로로 사용함

<도커 호스트 볼륨 설명>

호스트에서 도커 이미지 저장

도커 이미지는 `insect_conda2.tar` 이름으로 제공된다. 다음의 순서로 도커 이미지를 저장한다.

- 호스트 컴퓨터에 도커를 설치한다.
- `insect_conda2.tar` 파일을 호스트 컴퓨터에 복사한다.
- "`docker load -i insect_conda2.tar`" 명령으로 도커 이미지를 다시 복원한다.

호스트에서 도커 이미지 실행

다음은 우분투에서 도커를 실행하는 방법이다. 리눅스 또는 OSX 계열은 해당 OS의 도커 매뉴얼을 참조한다.(명령어는 차이가 없다)

도커 컨테이너 이름은 `insect_conda2:0.0` 이다.

호스트에서 컨테이너 실행

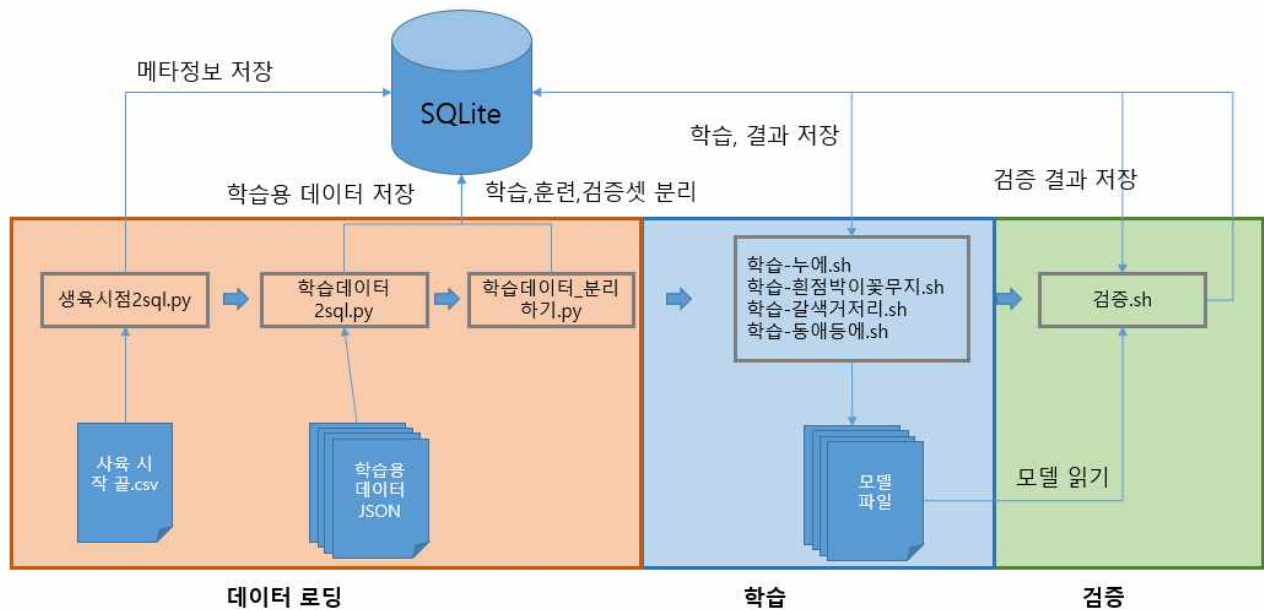
도커에서 명령어를 사용해서 진행할 것이기 때문에 `-t`, `-i` 옵션을 사용한다. 디스크 공유를 위해 `-v` 옵션을 사용한다.

```
# docker run -ti -v /usr/local/insect:/insect -e LC_ALL=C.UTF-8 insect_conda2:0.0
```

위 명령을 실행하면 입출력이 가능한 도커 콘솔창이 열린다. 이 창에서 작업을 순차적으로 진행하면 된다.

2-2 실행 프로그램

실행 프로그램은 다음과 같이 구성되어 있다. 화살표 방향으로 진행하면서 하나씩 실행한다.



생육시점 2sql.py

Note: 도커 이미지에 저장된 모델을 이용해 평가용 데이터셋을 검증하려면 이 단계는 건너뛰다.

학습에 사용할 json 파일의 메타정보를 저장한 csv 파일의 내용을 sqlite db 에 저장하는 작업이다. csv 파일의 이름은 "사육 시작 끝.csv"이다.

json 파일은 현재 곤충 이름(누에, 흰점박이꽃무지, ..) 및 상태(알, 애벌레, ..) 디렉토리에 저장되어 있기 때문에 디렉토리 이름으로 분류가 가능하다.

현재 json 파일에는 데이터 측정 시점은 존재하지만 상태 시작 시점(부화일, 애벌레 변태일, ..)은 존재하지 않으며 이 정보들은 csv 파일에 저장되어 있다. 따라서 학습에 용이하게 csv 파일의 내용을 미리 sqlite 테이블에 옮기는 작업을 해준다.

path 파라미터는 sqlite DB 를 만들거나 업데이트할 경로명이다. 도커의 호스트 폴더를 이용하면 호스트 파일 시스템에 남길 수 있다.

```
# python3 ./생육시점 2sql.py --path /insect/src
```

만약 도커이미지에 sqlite DB 를 만들 경우에는 path 파라미터를 생략한다.

Note : path 파라미터를 생략하면 도커에 만들어진 SQLite DB 는 도커 종료시 사라진다.


```

root@ubuntu:/usr/local/src/study/insect# docker run -ti -e LC_ALL=C.UTF-8 -v /usr/local/src/study/insect:/insect insect_conda2:0.0
no change      /root/miniconda3/condabin/conda
no change      /root/miniconda3/bin/conda
no change      /root/miniconda3/bin/conda-env
no change      /root/miniconda3/bin/activate
no change      /root/miniconda3/bin/deactivate
no change      /root/miniconda3/etc/profile.d/conda.sh
no change      /root/miniconda3/etc/fish/conf.d/conda.fish
no change      /root/miniconda3/shell/condabin/Conda.psm1
no change      /root/miniconda3/shell/condabin/conda-hook.ps1
no change      /root/miniconda3/lib/python3.9/site-packages/xontrib/conda.xsh
no change      /root/miniconda3/etc/profile.d/conda.csh
no change      /root/.bashrc
No action taken.
(py_3_8_11) root@0eff865fa95f:/usr/local/src/insect/src# python3 ./생육시점2sql.py --path /insect/src
에벌레 replace to:에벌레
흰점박이꽃무지 replace to:흰점박이꽃무지
insert into meta_06_1 values('실화상', '누에', '알', '고해상도', '20211009', '20211014')
insert into meta_06_1 values('실화상', '누에', '에벌레', '2번카메라', '20211103', '20211103')
insert into meta_06_1 values('실화상', '누에', '에벌레', '고해상도', '20210909', '20210914')
insert into meta_06_1 values('실화상', '누에', '에벌레', '고해상도', '20211104', '20211104')

.....
중략
.....

insert into meta_06_1 values('실화상', '동애등에', '성충', '2번카메라', '20211201', '20211207')
insert into meta_06_1 values('실화상', '동애등에', '성충', '3번카메라', '20211201', '20211207')
insert into meta_06_1 values('실화상', '동애등에', '성충', '4번카메라', '20211201', '20211207')
update meta_06_1 set insect = '흰점박이꽃무지' where insect = '흰점박이꽃무지'
Total :곤충 별태시점 메타 데이터 136 rows 생성

```

<도커 생육시점 2.sql.py 실행화면>

Note : 이 작업은 최초 1 번만 진행하며 "사육 시작 끝.csv" 파일의 내용을 sqlite(smart_insect_farm)의 meta_06_1 테이블에 옮겨넣는다.

학습데이터 2sql.py

Note: 도커 이미지에 저장된 모델을 이용해 평가용 데이터셋을 검증하려면 이 단계는 건너뛰다.

학습에 사용할 json 파일들을 읽어서 sqlite 테이블에 저장한다. json 파일의 개수가 너무 많기 때문에 json 파일을 직접 이용하면 데이터 로딩 속도가 많이 느려진다. 따라서 하이퍼 파라미터 변경 등 새롭게 학습을 진행할 필요가 발생한 경우 신속한 작업을 위해 sqlite 테이블로 저장하는 작업을 미리 한다.

테이블 개요

1. 곤충 종류에 따른 4 개의 테이블 작성 (누에,흰점박이꽃무지, 갈색거저리,동애등에)
2. 곤충의 상태(알,번데기,에벌레,성충) 분류 컬럼사용 -> 상태별 쿼리 가능
3. 생육시점 2sql 프로그램에서 만든 meta_06_1 테이블을 이용해 json 데이터의 센서데이터 측정 시간을 경과시간(시계열 데이터)로 변환
4. 테이블 이름 : insect_1_main(누에), insect_2_main(흰점박이꽃무지), insect_3_main(갈색거저리), insect_4_main(동애등에)

곤충 코드

누에 : 1, 흰점박이꽃무지 : 2, 갈색거저리 : 3, 동애등에 : 4

곤충 상태코드

알 : 1, 애벌레 : 2, 번데기 : 3, 성충 : 4

path 파라미터는 sqlite DB 를 만들거나 업데이트할 경로명이다. 도커의 호스트 폴더를 이용하면 호스트 파일 시스템에 남길 수 있다.

```
python3 ./학습데이터 2sql.py --path /insect/src
```

두가지 경우 모두 결과값을 저장하는 sqlite DB 는 호스트 폴더에 생성된다. 만약 도커이미지에 sqlite DB 를 만들 경우에는 path 파라미터를 생략한다.

Note : path 파라미터를 생략하면 도커에 만들어진 SQLite DB 는 도커 종료시 사라진다.

```
(py_3_8_11) root@0eff865fa95f:/usr/local/src/insect/src# python3 ./학습데이터2sql.py --path /insect/src

100개 변환
200개 변환

.....
종막
.....

122000개 변환
122100개 변환

{'total': 91505, '누에': [500, 4000, 15000, 3000], '흰점박이꽃무지': [500, 1000, 14440, 6958], '갈색거저리': [500, 3000, 12654, 7000], '동애등에': [500, 3000, 12451, 7002]}
{'total': 77384, '누에': [500, 4000, 14968, 3000], '흰점박이꽃무지': [500, 1000, 4700, 4149], '갈색거저리': [500, 3000, 11595, 7000], '동애등에': [499, 3000, 11988, 6985]}
{'total': 14121, '누에': [0, 0, 0, 0], '번데기': [0, 0, 0, 0], '애벌레': [0, 0, 0, 32, 0], '성충': [0, 0, 0, 0, 0]}, '흰점박이꽃무지': {'알': [0, 0, 0, 0, 0], '번
===== 종개 =====
종 작업 시도 json 파일 : 91505
누에 json 파일: 500
누에 번데기json 파일: 4000
누에 애벌레json 파일: 15000
누에 성충json 파일: 3000
누에 json 합계:22500
.....
종막
.....
동애등에 json 파일: 500
동애등에 번데기json 파일: 3000
동애등에 애벌레json 파일: 12451
동애등에 성충json 파일: 7002
동애등에 json 합계:22953
종 sql 변환 성공 갯수 : 77384
누에 합: 500
누에 번데기: 4000
누에 애벌레: 14968
누에 성충: 3000
누에 합계:22468
.....
종막
.....
동애등에 알: 499
동애등에 번데기: 3000
동애등에 애벌레: 11988
동애등에 성충: 6985
동애등에 합계:22472
곤충 sql 변환 총 합계:77384

변환 실패 파일 json 파일 : 14121
.....
종막
.....
```

<도커 학습데이터 2sql.py 실행화면>

Note : 이 작업은 최초 1 번만 진행한다. 다음과 같은 테이블이 만들어진다. 만약 테이블이 이미 존재하면 내용을 지우고 새롭게 만든다.

테이블 이름	내용
insect_1_main	누에
insect_2_main	흰점박이꽃무지
insect_3_main	갈색거저리
insect_4_main	동애등애

학습데이터_분리하기.py

Note: 도커 이미지에 저장된 모델을 이용해 평가용 데이터셋을 검증하려면 이 단계는 건너뛰는다.

이전 작업에서 SQLite 에 저장한 학습용 데이터를 학습용:테스트용:검증용으로 8:1:1 비율로 분리한다. 이 과정에서 라벨링 데이터 값이 0 인 데이터는 제외한다.

데이터 분리 작업 결과는 data_history 테이블에 transaction 으로 저장한다. 다음 공식이 성립한다.

$$\text{total} = \text{train_cnt} + \text{test_cnt} + \text{validation_cnt} + \text{drop_cnt}$$

data_history 테이블 컬럼	내용
train_cnt	학습용 데이터 숫자
test_cnt	테스트 데이터 숫자
validation_cnt	검증용데이터 숫자
total	원본 데이터 숫자
drop_cnt	라벨링값이 0 인 데이터 숫자

분리한 데이터는 다음 테이블 및 csv 파일로 저장한다.

테이블	csv 파일	내용
insect_1_main_len_train	insect_1_main_len_train.csv	누에 길이 훈련용
insect_1_main_len_test	insect_1_main_len_test.csv	누에 길이 테스트용
insect_1_main_len_validation	insect_1_main_len_validation.csv	누에 길이 검증용
insect_2_main_len_train	insect_2_main_len_train.csv	흰점박이꽃무지 길이 훈련용

insect_2_main_len_test	insect_2_main_len_test.csv	흰점박이꽃무지 길이 테스트용
insect_2_main_len_validation	insect_2_main_len_validation.csv	흰점박이꽃무지 길이 검증용
insect_3_main_len_train	insect_3_main_len_train.csv	갈색거저리 길이 훈련용
insect_3_main_len_test	insect_3_main_len_test.csv	갈색거저리 길이 테스트용
insect_3_main_len_validation	insect_3_main_len_validation.csv	갈색거저리 길이 검증용
insect_4_main_len_train	insect_4_main_len_train.csv	동애등에 길이 훈련용
insect_4_main_len_test	insect_4_main_len_test.csv	동애등에 길이 테스트용
insect_4_main_len_validation	insect_4_main_len_validation.csv	동애등에 길이 검증용
insect_1_main_weight_train	insect_1_main_weight_train.csv	누에 무게 훈련용
insect_1_main_weight_test	insect_1_main_weight_test.csv	누에 무게 테스트용
insect_1_main_weight_validation	insect_1_main_weight_validation.csv	누에 무게 검증용
insect_2_main_weight_train	insect_2_main_weight_train.csv	흰점박이꽃무지 무게 훈련용
insect_2_main_weight_test	insect_2_main_weight_test.csv	흰점박이꽃무지 무게 테스트용
insect_2_main_weight_validation	insect_2_main_weight_validation.csv	흰점박이꽃무지 무게 검증용
insect_3_main_weight_train	insect_3_main_weight_train.csv	갈색거저리 무게 훈련용
insect_3_main_weight_test	insect_3_main_weight_test.csv	갈색거저리 무게 테스트용
insect_3_main_len_validation	insect_3_main_len_validation.csv	갈색거저리 무게 검증용
insect_4_main_weight_train	insect_4_main_weight_train.csv	동애등에 무게 훈련용
insect_4_main_weight_test	insect_4_main_weight_test.csv	동애등에 무게 테스트용
insect_4_main_weight_validation	insect_4_main_weight_validation.csv	동애등에 무게 검증용

62 `select * from data_history order by serial;`

Execute SQL Stop Query

serial	dt	insect	insect_state	insect_type	train_cnt	test_cnt	validation_cnt	total	drop_cnt
1	2022-01-25 05:17:16	누에	애벌레	길이	7452	932	932	14968	5652
2	2022-01-25 05:17:16	누에	번데기	길이	3200	400	400	4000	0
3	2022-01-25 05:17:16	누에	성충	길이	2400	300	300	3000	0
4	2022-01-25 05:17:16	흰점박이꽃무지	알	길이	324	41	41	500	94
5	2022-01-25 05:17:17	흰점박이꽃무지	애벌레	길이	3760	470	470	4700	0
6	2022-01-25 05:17:17	흰점박이꽃무지	번데기	길이	797	100	100	1000	3
7	2022-01-25 05:17:17	흰점박이꽃무지	성충	길이	1770	221	222	4149	1936
8	2022-01-25 05:17:17	갈색거저리	알	길이	400	50	50	500	0
9	2022-01-25 05:17:17	갈색거저리	번데기	길이	2000	250	250	3000	500
10	2022-01-25 05:17:17	갈색거저리	성충	길이	4102	513	513	7000	1872
11	2022-01-25 05:17:17	동애등에	알	길이	260	33	33	499	173
12	2022-01-25 05:17:18	동애등에	애벌레	길이	7404	926	926	11988	2732
13	2022-01-25 05:17:18	동애등에	번데기	길이	2400	300	300	3000	0
14	2022-01-25 05:17:18	동애등에	성충	길이	4079	510	510	6985	1886
15	2022-01-25 05:17:18	누에	번데기	무게	3200	400	400	4000	0
16	2022-01-25 05:17:18	누에	성충	무게	560	70	70	3000	2300
17	2022-01-25 05:17:19	흰점박이꽃무지	애벌레	무게	3760	470	470	4700	0
18	2022-01-25 05:17:19	흰점박이꽃무지	번데기	무게	797	100	100	1000	3
19	2022-01-25 05:17:19	흰점박이꽃무지	성충	무게	1086	136	136	4149	2791
20	2022-01-25 05:17:19	갈색거저리	성충	무게	2722	340	341	7000	3597

<데이터분리 작업 저장 결과>

학습-(곤충).sh

Note: 도커 이미지에 저장된 모델을 이용해 평가용 데이터셋을 검증하려면 이 단계는 건너뛴다.

데이터 적재가 완료되면 학습을 진행할 수 있다. 학습 프로그램은 "학습하기.py" 파일이다. 총 64 개의 모델을 만들려면 명령어를 여러 번 입력해야 하기 때문에 곤충별로 학습을 진행할 수 있도록 만든 배시 스크립트 파일을 이용하면 편리하다.

스크립트 파라미터에는 도커에서 공유한 디렉토리를 사용한다. 만약 생략하면 도커에 저장된 sqlite 테이블을 이용하며 결과 모델 역시 도커 이미지 디렉토리에 저장한다..

Note : path 파라미터를 생략하면 도커에 만들어진 SQLite DB 및 만들어진 모델들은 도커 종료시 사라진다.

누에 학습

```
bash ./학습_누에.sh /insect/src
```

```
===== [동애등에] [월] [같이] 학습용 데이터통계 =====
학습용 데이터 갯수:260   컬럼수:18
테스트 데이터 갯수:33   컬럼수:18
===== Linear Regression 모델 생성 =====
Linear Regression 평균절대값오차: 3.4895688610252933e-06
===== Random Forest 모델 생성 =====
Random Forest 평균절대값오차: 4.01631313131376e-05
===== 선형 SVM 모델 생성 =====
C:\Users\spypi\anaconda3\envs\py_3_8_11\lib\site-packages\sklearn\svm\_base.py:985: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
warnings.warn("Liblinear failed to converge, increase ")
선형 SVM 평균절대값오차: 0.0003770000000000246

===== [동애등에] [애벌레] [같이] 학습용 데이터통계 =====
학습용 데이터 갯수:7404   컬럼수:18
테스트 데이터 갯수:926   컬럼수:18
===== Linear Regression 모델 생성 =====
Linear Regression 평균절대값오차: 14.392788967795347
===== Random Forest 모델 생성 =====
Random Forest 평균절대값오차: 0.05442385852638762
===== 선형 SVM 모델 생성 =====
C:\Users\spypi\anaconda3\envs\py_3_8_11\lib\site-packages\sklearn\svm\_base.py:985: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
warnings.warn("Liblinear failed to converge, increase ")
선형 SVM 평균절대값오차: 3.9936036965442763

.....생략 .....
```

<도커 학습데이터 학습_동애등에.sh 실행화면>

흰색점박이꽃무지 학습

```
bash ./학습_흰색점박이꽃무지.sh /insect/src
```

갈색거저리 학습

```
bash ./학습_갈색거저리.sh /insect/src
```

동애등에 학습

```
bash ./학습_동애등에.sh /insect/src
```

Note: 학습에는 훈련용데이터 및 테스트용 데이터를 사용한다.

훈련용으로는 모델을 생성하며 테스트용으로는 만든 모델의 평가(MAE 값을 계산)하는데 사용한다. 테스트셋으로 평가한 결과는 train_hostory 테이블에 저장된다.

학습용 데이터 분리 과정에서 유효한 데이터가 없는 경우에는 모델이 생성되지 않는다.

검증.sh

검증은 학습과정에서 저장해둔 8:1:1 의 마지막 1 분량을 이용한다.

검증용 데이터는 다음 테이블에 있다.

테이블 명	내용	비고
insect_1_main_len_validation	누에 길이 검증용	
insect_1_main_weight_validation	누에 무게 검증용	
insect_2_main_len_validation	흰점박이꽃무지 길이 검증용	
insect_2_main_weight_validation	흰점박이꽃무지 무게 검증용	
insect_3_main_len_validation	갈색거저리 길이 검증용	
insect_3_main_weight_validation	갈색거저리 무게 검증용	
insect_4_main_len_validation	동애등에 길이 검증용	
insect_4_main_weight_validation	동애등에 무게 검증용	

검증 작업은 최대 64 개 모델(학습데이터가 없는 경우에는 모델 생성 안됨)에 대해서 수행한다. 만약 특정 모델만 테스트하려면 배시 스크립트 대신 "검증하기.py"를 직접 사용한다. 검증 작업은 검증셋 데이터를 이용하며 검증셋의 라벨링 데이터(길이,무게)와 모델의 예측값과의 차이(MAE)를 계산한다.

스크립트 파라미터에는 도커에서 공유한 디렉토리를 사용한다. 만약 생략하면 도커에 저장된 sqlite 테이블을 이용하며 결과 모델 역시 도커 이미지 디렉토리에 저장한다..

Note : path 파라미터를 생략하면 도커에 만들어진 SQLite DB 및 만들어진 모델들은 도커 종료시 사라진다.

1. 호스트 공유 디렉토리의 모델 및 데이터셋을 이용할 경우

모든 모델에 대해서 검증을 시도할 경우에는 배시 스크립트를 이용한다.

```
bash ./검증.sh /insect/src
```

개별 모델 검증은 파이썬 프로그램을 따로 실행한다. (예: 누에 성충 길이를 random forest 모델을 이용해 예측)


```
python ./검증하기.py --insect 누에--state 성충 --type 길이 --model random --path
/insect/src
```

2. 도커 이미지의 모델 및 데이터셋을 이용할 경우

모든 모델에 대해서 검증을 시도할 경우에는 파라미터를 생략하고 배시 스크립트를 이용한다.

```
bash ./검증.sh
```

개별 모델 검증은 파이썬 프로그램을 따로 실행한다. (예: 누에 성충 길이를 random forest 모델을 이용해 예측)

```
python ./검증하기.py --insect 누에--state 성충 --type 길이 --model random
```

```
===== 모델 로딩 =====
모델 파일 Error:..\model\누에_알_길이_linearsvm.pkl
===== 모델 로딩 =====
모델 파일 Error:..\model\누에_알_길이_randomforest.pkl
===== 모델 로딩 =====
모델 이름:..\model\누에_애벌레_길이_linearsvm.pkl
검증 데이터 세트 [932] 로딩
==== 모델: ..\model\누에_애벌레_길이_linearsvm.pkl
검증 데이터 세트 평균절대값오차: 3.00
===== 모델 로딩 =====
모델 이름:..\model\누에_애벌레_길이_randomforest.pkl
검증 데이터 세트 [932] 로딩
==== 모델: ..\model\누에_애벌레_길이_randomforest.pkl
검증 데이터 세트 평균절대값오차: 0.00
===== 모델 로딩 =====
모델 이름:..\model\누에_번데기_길이_linearsvm.pkl
검증 데이터 세트 [400] 로딩
==== 모델: ..\model\누에_번데기_길이_linearsvm.pkl
검증 데이터 세트 평균절대값오차: 2.82
..... 종료.....

===== 모델 로딩 =====
모델 이름:..\model\갈색거저리_성충_무게_randomforest.pkl
검증 데이터 세트 [341] 로딩
==== 모델: ..\model\갈색거저리_성충_무게_randomforest.pkl
검증 데이터 세트 평균절대값오차: 0.00
===== 모델 로딩 =====
모델 파일 Error:..\model\똥애등에_성충_무게_linearsvm.pkl
===== 모델 로딩 =====
모델 파일 Error:..\model\똥애등에_성충_무게_randomforest.pkl
```

< 도커 학습데이터 검증.sh 실행화면 >

Note: 검증결과는 validation_history 테이블에 저장된다.

validation_history 테이블 컬럼	내용
model_name	검증 모델명
validation_cnt	검증용 데이터 숫자
mean	검증용 데이터 라벨링(길이, 무게) 평균
MAE	평균절대오차값
MAPE	평균절대오차 퍼센트(100 을 곱하지 않은 값)

69 select * from validation_history;

serial	dt	insect	insect_state	insect_type	model_name	validation_cnt	mean	MAE	MAPE
1	2022-01-25 07:56:05	누에	애벌레	길이	linearsvm	932	16.43	3	0.18
2	2022-01-25 07:56:06	누에	애벌레	길이	randomforest	932	16.43	0	0
3	2022-01-25 07:56:08	누에	번데기	길이	linearsvm	400	34.41	2.82	0.09
4	2022-01-25 07:56:09	누에	번데기	길이	randomforest	400	34.41	0.42	0.01
5	2022-01-25 07:56:11	누에	성충	길이	linearsvm	300	7.41	2.55	0.97
6	2022-01-25 07:56:12	누에	성충	길이	randomforest	300	7.41	0	0
7	2022-01-25 07:56:14	흰점박이꽃무지	알	길이	linearsvm	41	3.54	1.54	0.14
8	2022-01-25 07:56:15	흰점박이꽃무지	알	길이	randomforest	41	3.54	0	0
9	2022-01-25 07:56:17	흰점박이꽃무지	애벌레	길이	linearsvm	470	28.06	2.16	0.08
10	2022-01-25 07:56:18	흰점박이꽃무지	애벌레	길이	randomforest	470	28.06	0	0
11	2022-01-25 07:56:20	흰점박이꽃무지	번데기	길이	linearsvm	100	22.58	0.58	0.02
12	2022-01-25 07:56:21	흰점박이꽃무지	번데기	길이	randomforest	100	22.58	0	0
13	2022-01-25 07:56:23	흰점박이꽃무지	성충	길이	linearsvm	222	15.29	11.26	0.5
14	2022-01-25 07:56:24	흰점박이꽃무지	성충	길이	randomforest	222	15.29	0.69	0.13
15	2022-01-25 07:56:26	갈색거저리	알	길이	linearsvm	50	1	0	0
16	2022-01-25 07:56:27	갈색거저리	알	길이	randomforest	50	1	0	0
17	2022-01-25 07:56:32	갈색거저리	번데기	길이	linearsvm	250	17.42	5.42	0.27
18	2022-01-25 07:56:33	갈색거저리	번데기	길이	randomforest	250	17.42	0.43	0.03
19	2022-01-25 07:56:35	갈색거저리	성충	길이	linearsvm	513	13.91	0.91	0.06
20	2022-01-25 07:56:37	갈색거저리	성충	길이	randomforest	513	13.91	0	0

<검증 데이터셋을 이용한 검증결과>

Note: 도커에 저장된 로그 파일과 업데이트된 SQLite 테이블 내용은 도커 재실행시 리셋된다. 따라서 필요한 경우 반드시 미리 로그파일, db 파일을 도커 종료 전에 복사해둔다.

모든 로그 파일을 호스트의 ~/data 디렉토리에 복사할 경우 다음과 같이 사용한다.

```
docker cp conda2/usr/local/src/study/insect/log/*.log ~/data/
```

데이터셋이 저장된 sqlite db 를 및 csv 파일을 호스트의 ~/data 디렉토리에 복사할 경우 다음과 같이 사용한다.

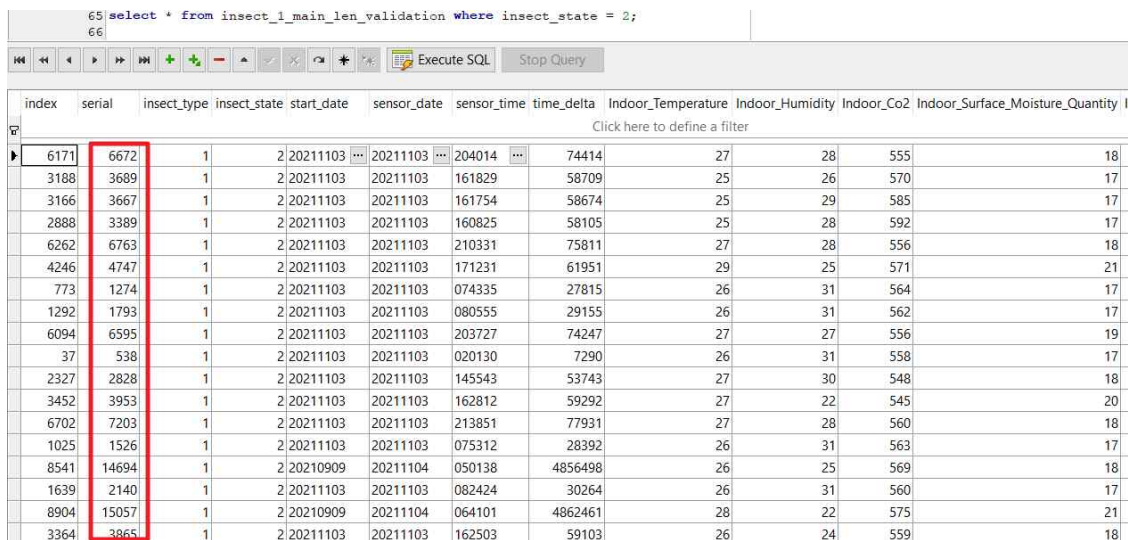
```
docker cp conda2/usr/local/src/study/insect/sql/*.db ~/data/
```

```
docker cp conda2/usr/local/src/study/insect/sql/*.csv ~/data/
```

생장예측.py

검증 데이터셋에서 특정 데이터를 추출해 생장모델에 대입하는 프로그램이다. 검증 데이터셋에서 추출한 검증 데이터의 센서 값을 자세하게 보여준다. 특정 데이터를 추출하기 위해서는 검증 데이터셋이 저장된 테이블 row 의 serial 컬럼 값을 알아야 한다. SQLite 도구를 이용해서 알 수 있다.

또는 csv 파일을 이용할 수 있다. 테이블 이름과 동일한 csv 파일을 텍스트 편집기로 열어 serial 값을 확인할 수 있다.



The screenshot shows a SQLite query interface. The query entered is: `select * from insect_i_main_len_validation where insect_state = 2;`. The results are displayed in a table with the following columns: index, serial, insect_type, insect_state, start_date, sensor_date, sensor_time, time_delta, Indoor_Temperature, Indoor_Humidity, Indoor_Co2, Indoor_Surface_Moisture_Quantity. The 'serial' column is highlighted with a red box, and the value '6672' is selected in the first row.

index	serial	insect_type	insect_state	start_date	sensor_date	sensor_time	time_delta	Indoor_Temperature	Indoor_Humidity	Indoor_Co2	Indoor_Surface_Moisture_Quantity
6171	6672	1	2	20211103	20211103	204014	74414	27	28	555	18
3188	3689	1	2	20211103	20211103	161829	58709	25	26	570	17
3166	3667	1	2	20211103	20211103	161754	58674	25	29	585	17
2888	3389	1	2	20211103	20211103	160825	58105	25	28	592	17
6262	6763	1	2	20211103	20211103	210331	75811	27	28	556	18
4246	4747	1	2	20211103	20211103	171231	61951	29	25	571	21
773	1274	1	2	20211103	20211103	074335	27815	26	31	564	17
1292	1793	1	2	20211103	20211103	080555	29155	26	31	562	17
6094	6595	1	2	20211103	20211103	203727	74247	27	27	556	19
37	538	1	2	20211103	20211103	020130	7290	26	31	558	17
2327	2828	1	2	20211103	20211103	145543	53743	27	30	548	18
3452	3953	1	2	20211103	20211103	162812	59292	27	22	545	20
6702	7203	1	2	20211103	20211103	213851	77931	27	28	560	18
1025	1526	1	2	20211103	20211103	075312	28392	26	31	563	17
8541	14694	1	2	20210909	20211104	050138	4856498	26	25	569	18
1639	2140	1	2	20211103	20211103	082424	30264	26	31	560	17
8904	15057	1	2	20210909	20211104	064101	4862461	28	22	575	21
3364	3865	1	2	20211103	20211103	162503	59103	26	24	559	18

<SQLite 도구를 이용한 누에 애벌레 검증데이터셋의 시리얼번호 추출 예 >

serial 값을 이용해 다음과 같이 생장예측 프로그램을 실행할 수 있다. 위에서 확인한 serial 값 중 하나를 선택한 다음 생장예측 프로그램을 돌려본다.

```

(py_3.8.11) C:\lsh\study\insect\final\src>python 생장예측.py --insect 누에 --state 애벌레 --type 길이 --idx 6672
===== 모델 로딩 =====
모델 로딩:..\model\누에 애벌레 길이_randomforest.pkl
모델 로딩:..\model\누에 애벌레 길이_linearsvm.pkl
검증 데이터 세트 [1] 로딩
==== 모델: randomforest
검증 데이터 세트 평균절대값오차: 0.00
==== 모델: svm
검증 데이터 세트 평균절대값오차: 3.00
곤충: 누에 단계: 애벌레
예측 대상: 길이
===== 입력값(센서 데이터) =====
단계 시작일: 20211103
측정 일: 20211103
측정 시간: 204014
경과 시간(초): 74414
실내 온도: 27.00
실내 습도: 28.00
실내 CO2농도: 555.00
곤충 표면 수문량: 18.00
실내 조도: 0.00
암모니아 농도: 775.00
산소 농도: 21.00
외부 풍속: 0.00
외부 온도: 3.00
외부 습도: 90.00
사육상자 온도: 24.00
사육상자 습도: 40.00
사육상자 CO2농도: 473.00
사육상자 매트 온도: 24.00
===== 사육 정보(관리 데이터) =====
곤충 길이: 16.00
곤충 무게: 0.00
===== AI 모델 출력값 =====
Random Firest 예측 길이: 16.00
Random Firest MAE: 0.00010 MAPE: 0.00001
선형 SVM 예측 길이: 19.00
선형 SVM MAE: 3.00090 MAPE: 0.18756

```

<누에 애벌레 검증데이터셋의 시리얼번호 6375 번 데이터 예측결과 >