머신러닝 활용 태양광 발전 예측

2021. 9

도시가스사업본부 기술연구소

< 과제 유무형 효과 >

- **전력중개사업 및 재생E 발전량 예측 제도 참여 가정 (모집 자원 태양광 20MW 가정)**
  - 재생E 발전량 예측 인센티브 배분은 획득 가능한 최대 인센티브의 50% 고객 지급 보장 (업계 표준)
  - 당사 자원은 오산, 광명 참여 (총 0.32MW)
- **유형 효과**
  - 전력중개사업 참여 시 연간 인센티브 9.9백만원 획득 및 사업 진입을 위한 초기 투자비 40백만원(추정) 절감
- **무형 효과**
  - 가상발전소(VPP)의 핵심 기술로 평가되는 재생E 발전량 예측 기술 확보 기반 마련
  - AI(머신러닝 등) 관련 기술 활용 경험을 통한 기술연구소 역량 배양 및 확대 적용 (에너지 수요 예측 등)

구분		자원 용량 (MW)	인센티브 (MWh, 백만원)					
			대상 발전량[1]	최대 인센티브[2]	인센티브 획득률	획득 인센티브	당사 배분[3]	고객 배분[4]
매출	**총 모집 자원**	**20.0**	**2,473.9**	**118.7**	**57.6%**	**68.4**	**9.9**	**58.4**
	당사 보유 자원	0.3	39.6	1.9	57.6%	1.1	1.1	0.0
	외부 자원	19.7	2,434.3	116.8	57.6%	67.3	8.8	58.4

1) 대상 발전량 : 자원 용량의 10% 이상 발전 시 발전량 합계
2) 최대 인센티브 : 최대로 획득할 수 있는 인센티브로 전체 시간의 오차율이 6% 이하인 (대상 발전량 × 4원/kWh)
3) 당사 배분 : 당사 보유 자원은 획득 인센티브의 100%, 외부 자원은 획득 인센티브에서 고객 배분을 차감 후 잔여 인센티브
4) 고객 배분 : 최대 인센티브의 50%

※ 기타 사항은 발전량 예측 자료 본문 참고

구분		금액	비고
예상 총투자비	계량기 교체비 (오산, 광명)	4	한전 계량기 → 전력거래소 계량기
	자원 모니터링 및 관리 시스템	40	모집자원 40개소 가정 (100만원/개소)
	발전량 예측 시스템 도입	40	남동발전 사례 (협력사 : 브이젠)
	합계	**84**	
절감액	**발전량 예측 시스템**	**- 40**	
예상 순투자비 (예상 총 투자비 – 절감액)		**44**	

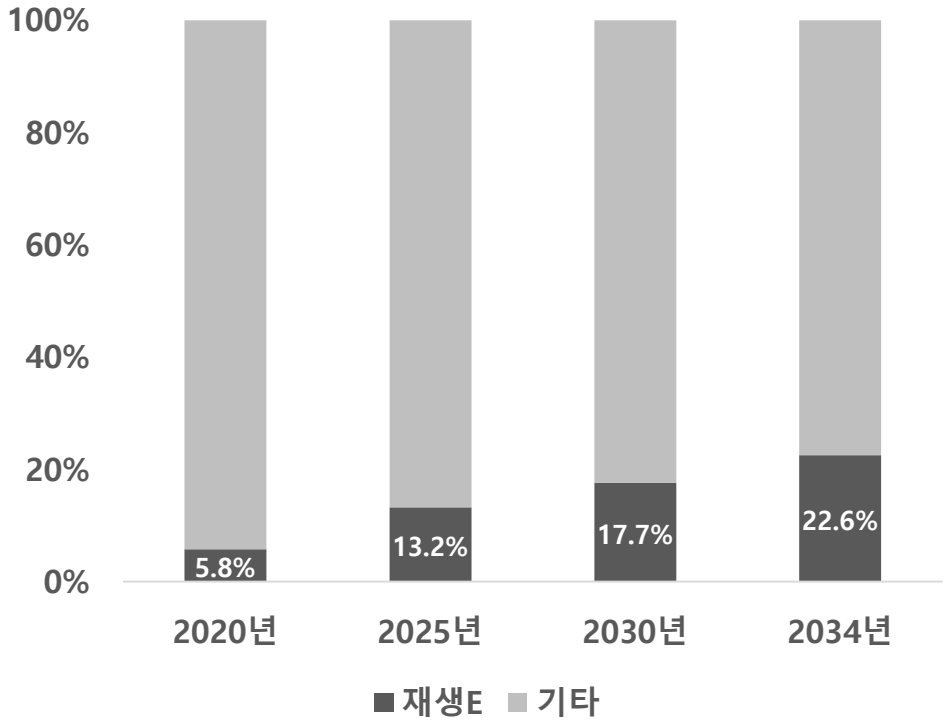
목 차

Ⅰ. 태양광 발전량 예측 필요성

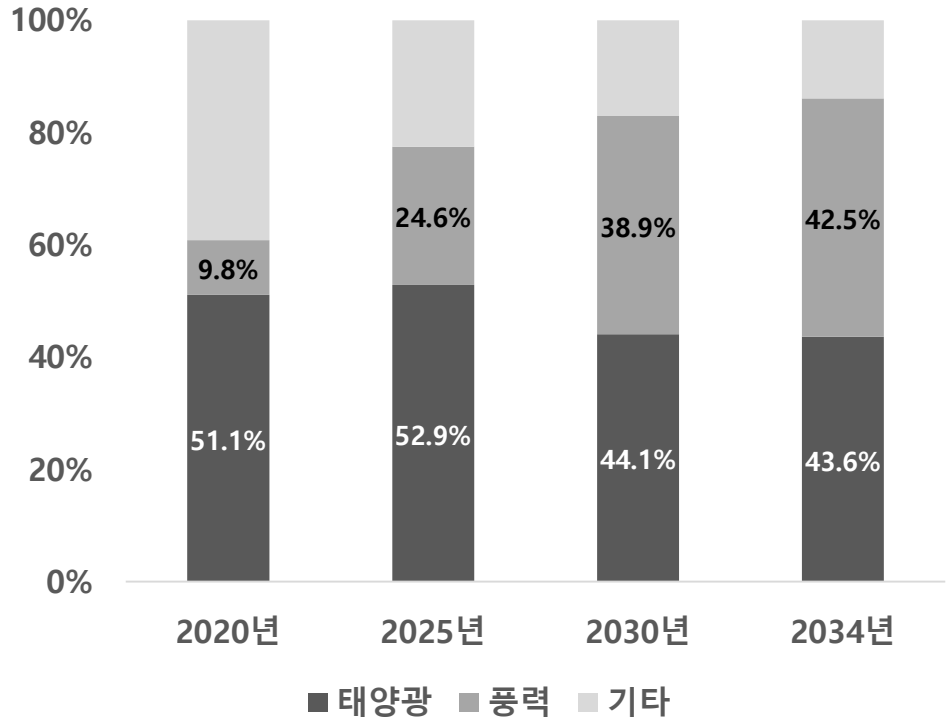
재생E 보급에 따른 계통 불안정 | 지역E 자립 모델

- **재생E 보급이 급격하게 증가함에 따라 태양광/풍력의 간헐성으로 인한 전력계통 불안정 가속화**
  - '34년 기준 재생E는 전체 발전량의 22.6%, 이 중 태양광/풍력이 약 86%를 차지할 것으로 전망
- **재생E의 간헐성 보완을 위한 대체수단(가스터빈 등)은 급격한 발전량 변화 감당에 한계 존재**
- **재생E 발전량 예측은 계통 안정 목적 외에 VPP, 지역E 자립 등 신규 사업모델에서 중요한 역할을 할 것으로 예상**

재생E 보급 전망



재생E 중 태양광/풍력 비율

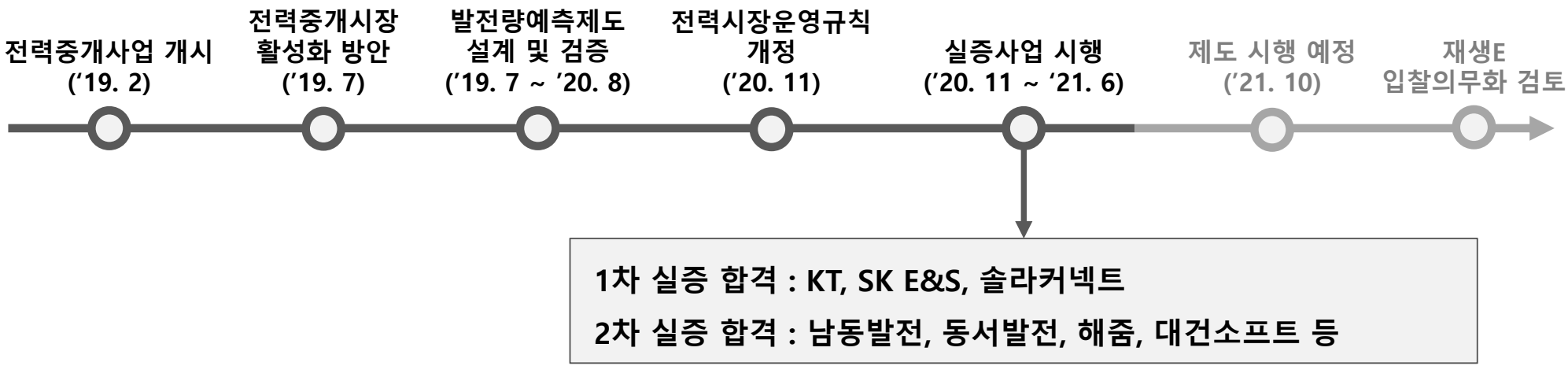


Samchully

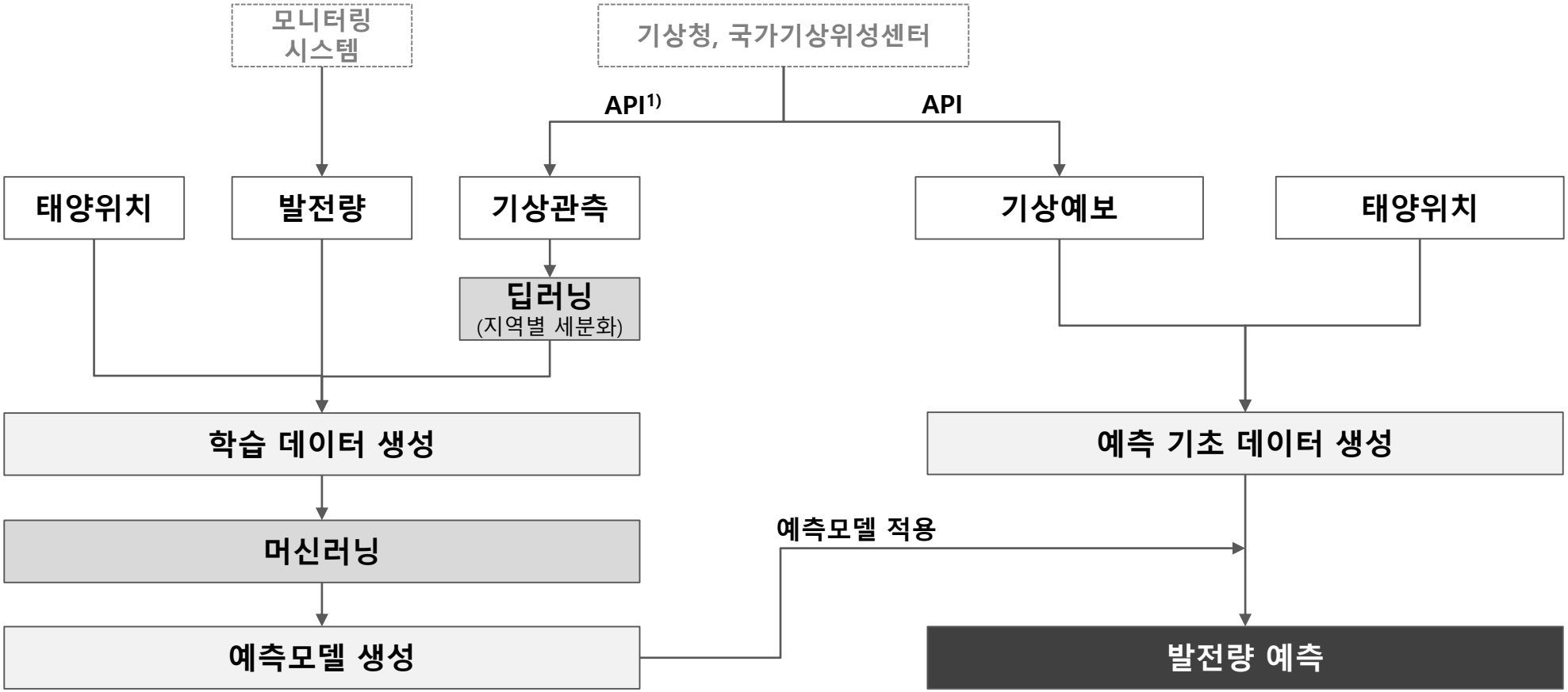
Ⅱ. 재생E 발전량 예측 제도

- 태양광/풍력 발전사업자 및 전력중개사업자를 대상으로 발전량 예측 정확도에 따라 인센티브 지급하는 제도
- 재생E 출력 변동성 완화를 통해 전력계통 안정성 향상 및 발전비용 감소 도모
- 향후 재생에너지 입찰 의무화 대비 사업자 역량 강화

구분	내용	
참여 대상	20MW를 초과하는 태양광/풍력을 발전원으로 하는 발전사업자 또는 전력중개사업자	
등록 요건	평균 예측오차율 10% 이상 (등록시험 1개월간), 3개월 평균 예측오차율 10% 초과 시 참여 대상 제외	
인센티브	예측 오차율 6% 이하	4원/kWh
	예측 오차율 6% 초과 ~ 8% 이하	3원/kWh



- 태양위치 계산, 기상관측 데이터, 발전실적을 결합한 학습데이터를 기반으로 머신러닝을 통해 예측모델 생성
- 예측하고자 하는 날에 예측모델을 적용하여 발전량 예측 시행



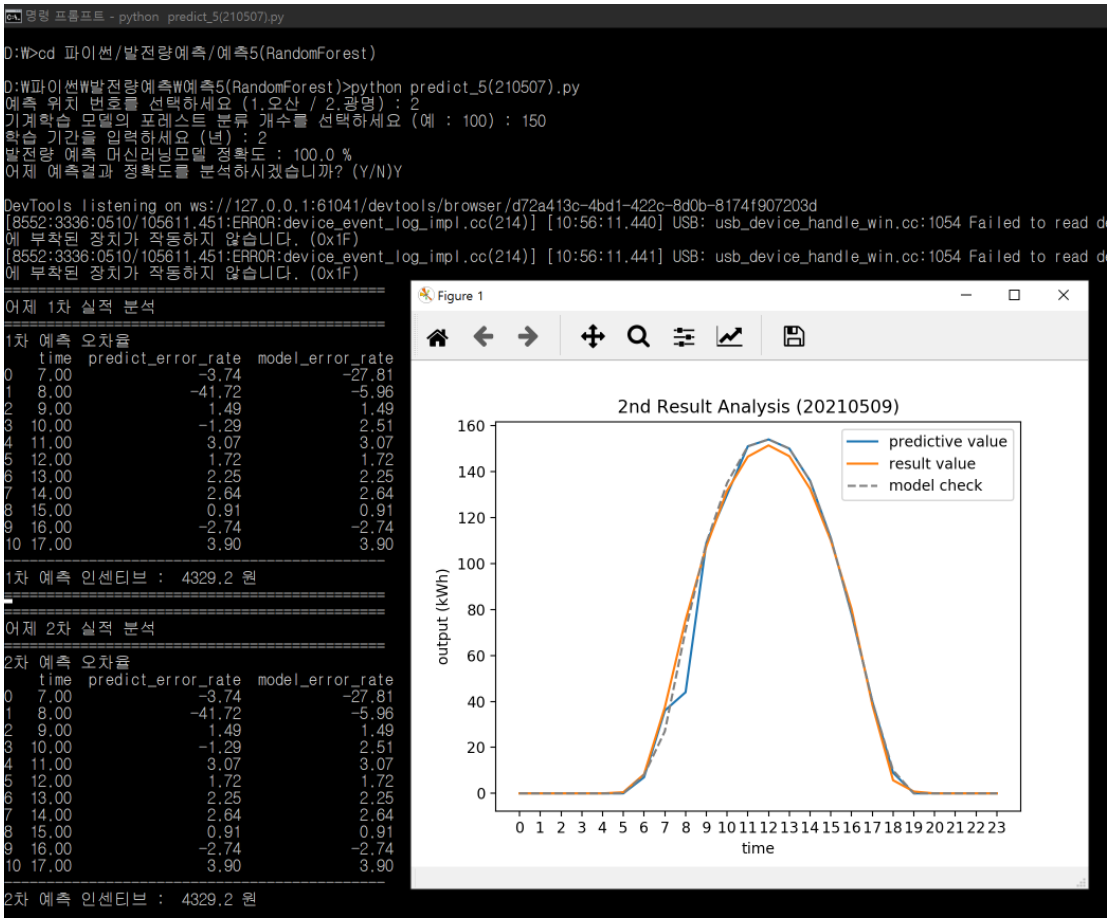
1) API (Application Programming Interface) : 일정 데이터를 얻기 위해 반복하여 수행하는 함수에 접근하기 위해 정의된 규칙

Ⅲ. 예측 알고리즘

학습모델 세부 내용

- **학습데이터(태양위치 + 기상데이터 + 발전량) 생성 후 머신러닝 수행 (파이썬 'Scikit-learn' 활용)**
  - 학습데이터 항목 : 태양고도, 태양고도 변화율, 태양방위각, 기온, 풍속, 습도, 운량, 강수량, 발전량
- **4가지 학습모델을 선정하여 학습 진행 및 모델 생성 → 검증기간 이후 예측 정확도 높은 알고리즘 선택**

구분	학습모델	설명
모델 1	다중선형회귀	▪ 선형회귀방정식 산출
모델 2	K-최근접이웃 (kNN)	▪ 주어진 데이터에서 가장 근접한 이웃값의 평균 추출
모델 3	랜덤 포레스트	▪ 다수의 결정트리를 랜덤하게 생성 ▪ 생성된 결정트리의 결과들을 결합하여 투표에 의해 최종 결과 산출
모델 4	서포트벡터머신 (SVM)	▪ 데이터 간 경계를 설정 ▪ 경계에 위치한 데이터를 벡터화하여 거리 계산 ▪ 비선형 함수를 적용하여 경계 설정 최적화



Samchully

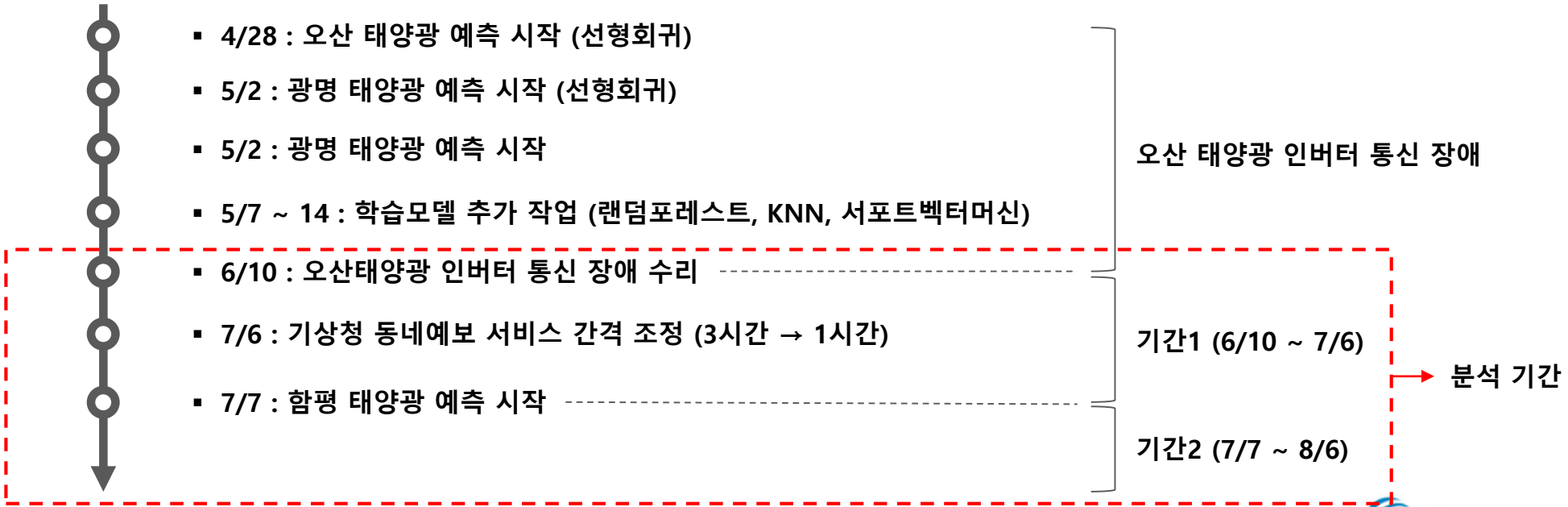
Ⅳ. 예측 시행 결과

- **집합자원 구성 시 자원들이 일정 지역에 집중되기 보다는 전국적으로 골고루 분포하는 것이 정확도 제고에 유리**
- **당사 보유 태양광 발전소 중 오산, 광명, 함평을 대상으로 예측을 시행하였으며 함평의 경우 지역별 배분 영향을 검토하기 위해 오산, 광명과 동일한 용량만 축소 반영**

사이트	지역	설비 용량	예측 용량	예측 시작일	비고
오산 기술연구소	경기 오산	147 kW	147 kW	'21. 4. 28	
광명 열병합사업단	경기 광명	173 kW	173 kW	'21. 5. 5	
함평 태양광	전남 함평	1,996 kW	320 kW	'21. 7. 7	용량 축소
합계		**2,556 kW**	**640 kW**		

- **최초 4/28에 예측을 시작하였으나 6/10 이전의 오산 태양광 인버터 통신 장애로 예측 결과는 6/10 이후만 분석**

  - 기상청 예보 서비스 간격 조정(7/6, 3시간→1시간)을 기준으로 기간1(6/10~7/6), 기간2(7/7~8/6)로 나누어 분석

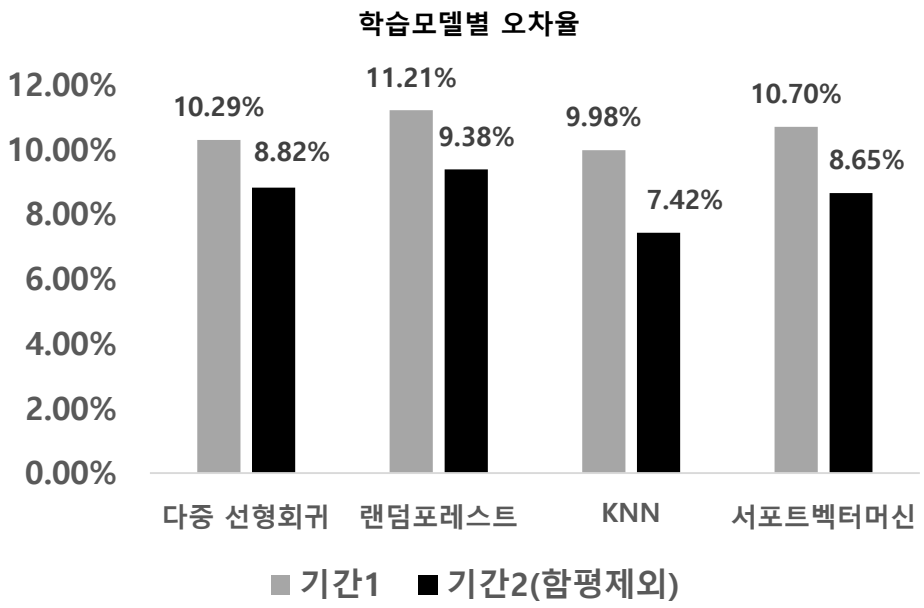


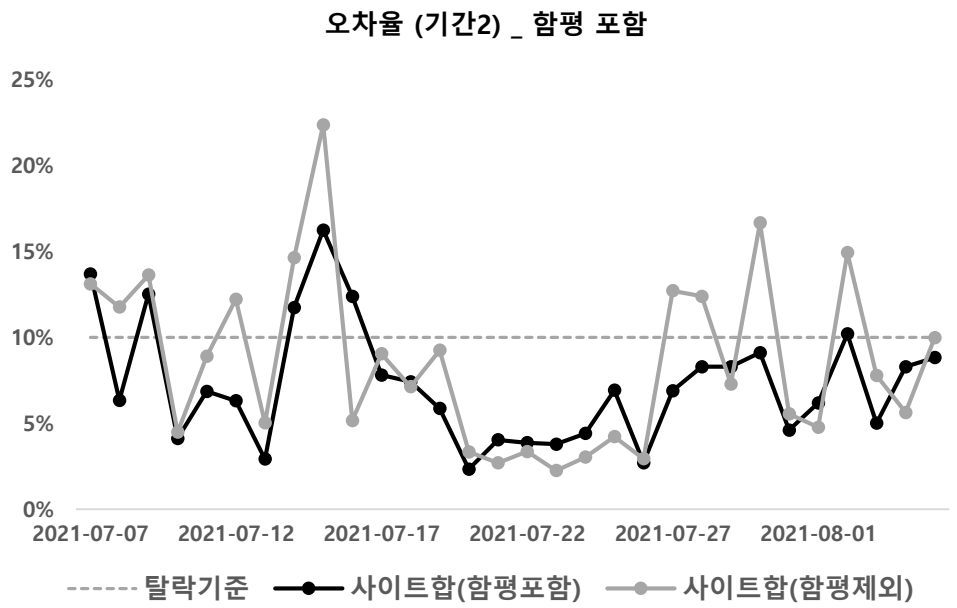
Ⅴ. 예측 결과

세부 예측 결과

- **기간2의 최종 오차율은 7.20%로 재생E 발전량예측제도 등록 기준 (10%)에 부합**
  - 학습모델 중 서포트벡터머신 기준(KNN의 오차율이 가장 낮지만 기상예보 적중 유무에 따른 신뢰도 문제 존재)
- **함평 제외 시 오차율이 8.65%로 자원이 넓은 지역에 분포되었을 경우 정확도가 높은 것으로 확인**
- **기상청 예보서비스가 세분화 된 기간2의 정확도가 기간1에 비해 향상 (오차율 : 10.70% → 8.65%)**

기간1		기간2 (함평 제외)		기간2 (함평 포함)
10.70%	기상예보 세분화 →	8.65%	함평 포함 →	7.20%

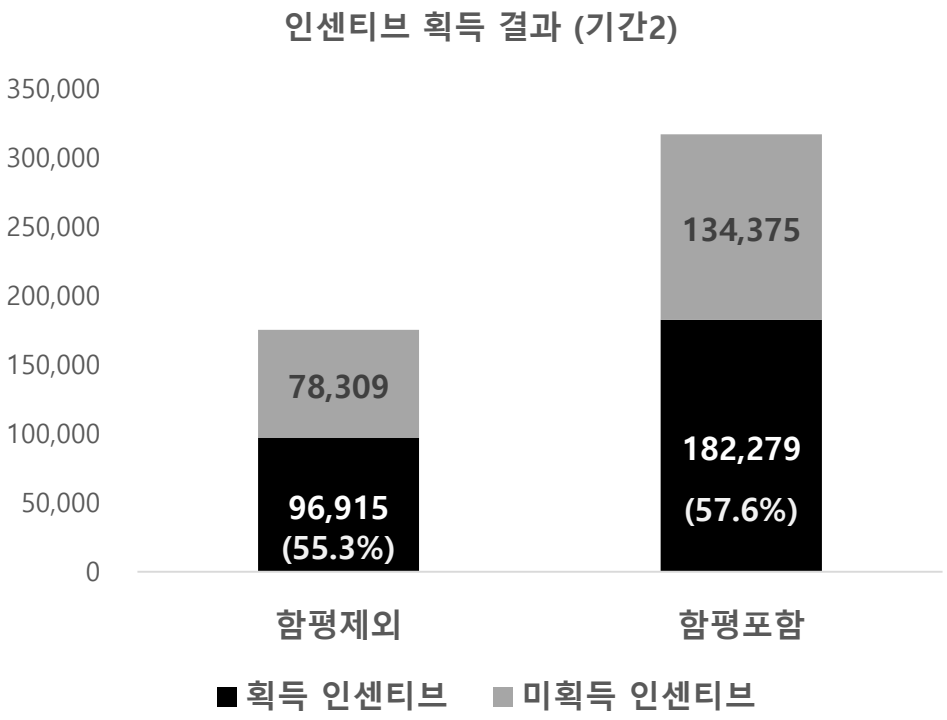




Samchully

Ⅴ. 예측 결과 _ 분석 및 보완점

- **분석 결과 오차율은 7.20%로 재생E 발전량예측제도 등록 기준에 부합하지만 인센티브 획득률은 57.6%로 저조**
  - 제도상 6% 초과 ~ 8% 이하의 오차율은 최고 4원/kWh 중 3원/kWh의 인센티브 획득 가능 (75%)
  - 발전량이 많은 시간대의 오차율이 증가할 수밖에 없는 오차율 계산 방식이 원인 (오차율 = 오차 ÷ 자원 용량)
- **실제 사업 수행을 위한 충분한 인센티브 획득을 위해서는 더 높은 정확도가 필요**
- **정확도 향상을 위해 운량 예보의 세분화, 기상 예보의 지역 편차 고려 전국단위의 자원 포트폴리오 구성 등 필요**



운량 데이터 신뢰도 증가를 통한 예측 정확도 향상 방안

기상 예보		기상 관측
4단계	정도 차이 →	10단계
동 단위 행정구역	지역 차이 →	기상관측소 (96개소)

추가 검토

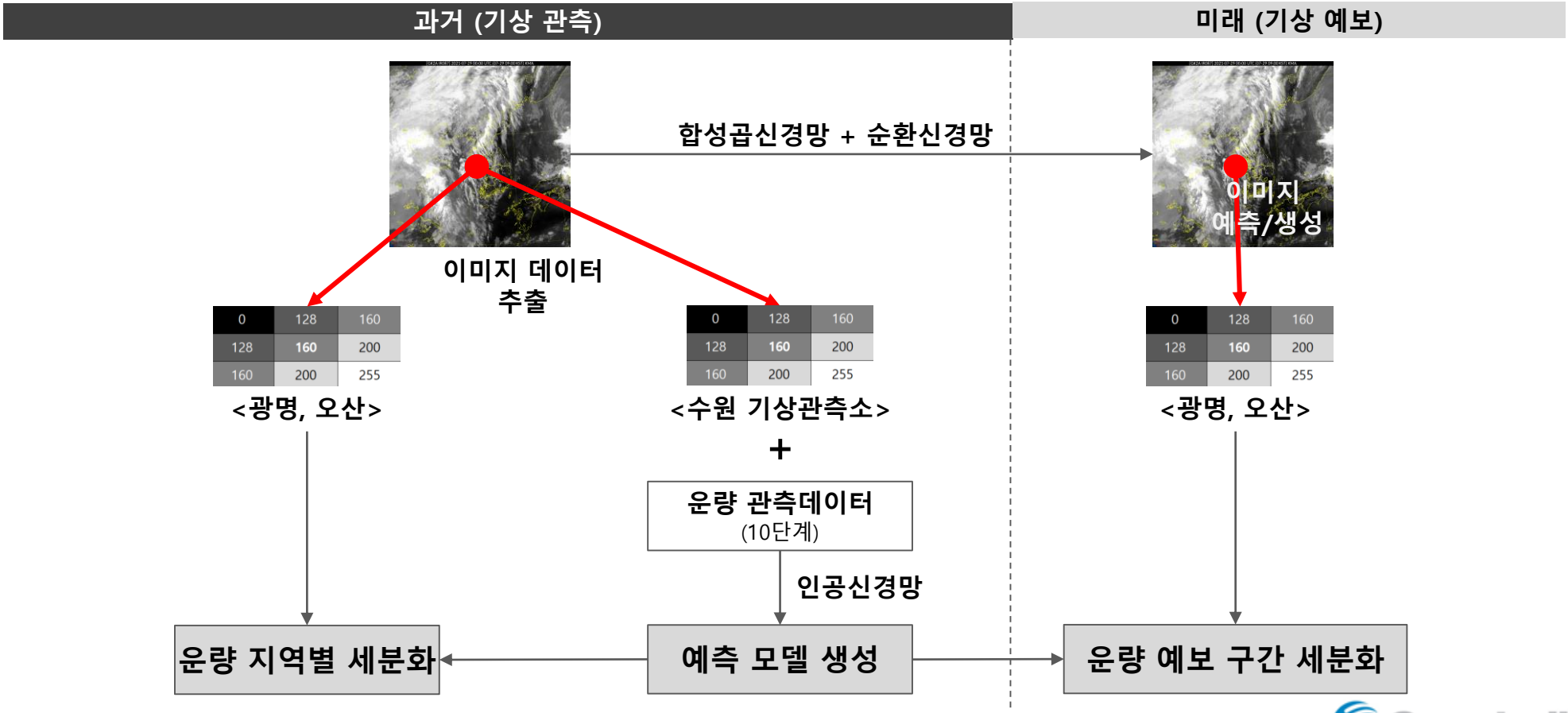
천리안2호 기상위성영상 활용 운량 데이터 세분화

1단계 : 지역 세분화	인공신경망을 이용한 지역별 운량 세분화
2단계 : 정도 세분화	합성곱신경망을 이용한 위성사진 예측

Samchully

VI. 정확도 향상 방안 _ 천리안2호 기상위성 사진 활용

- **예측 정확도 제고를 위해 운량데이터를 지역별(관측소 지역 → 예측 지역), 구간별(4단계 → 10단계)로 세분화**
- **머신러닝의 난이도와 필요 시스템 사양 등을 고려하여 단계별로 추진**
  - 1단계 (지역별 세분화) : 이미지 데이터 추출 및 인공신경망 활용한 운량 예측
  - 2단계 (구간별 세분화) : 고급 딥러닝 기법 활용 기상 위성 이미지 예측 및 생성 → 향후 추진 계획



Samchully

VI. 정확도 향상 방안 _ 운량 지역별 세분화

- 국가기상위성센터(nmsc.kma.go.kr)에서 제공하는 천리안2호 기상위성 적외(8.7㎛) 영상을 기반으로 시행
- 이미지 처리 → 이미지데이터 추출 → 학습데이터 구성 → 딥러닝(인공신경망) → 예측 기초데이터 구성 → 예측
- 구글의 머신러닝 플랫폼인 텐서플로우(TensorFlow)의 케라스(keras) 모듈을 사용하여 인공신경망 구성

이미지 처리
(예시 : 수원 기상관측소)

RGB → 흑백
(900×900×3) → (900×900)

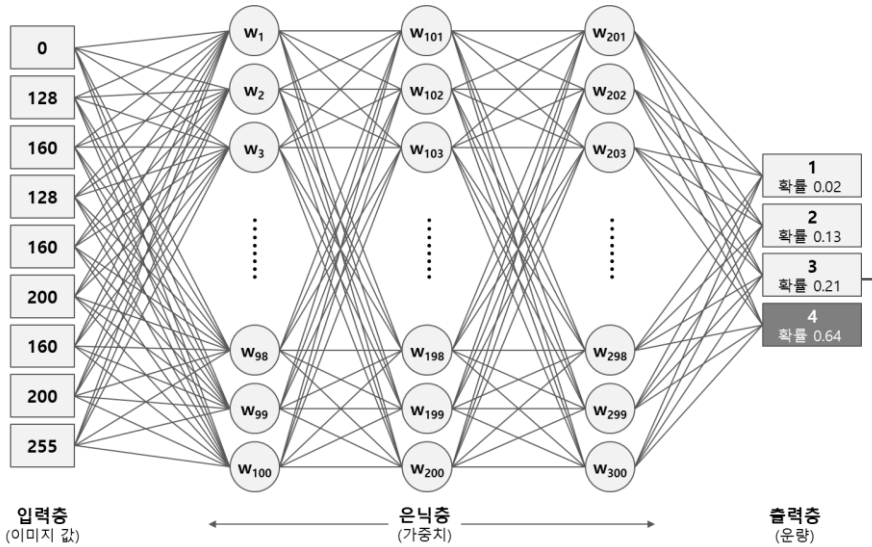
위도경도 → 픽셀값
(37.26, 126.98) → (490, 493)

지역 이미지데이터 추출
(3×3)

이미지데이터 행렬 → 벡터
(3×3) → (9×1)

학습데이터 구성

이미지데이터 벡터 + 운량
(9×1) → (10×1)



딥러닝 (인공신경망) → 예측모델 생성

예측 기초데이터 구성
(예시 : 오산 기술연구소)

위도경도 → 픽셀값
(37.18, 127.03) → (494, 495)

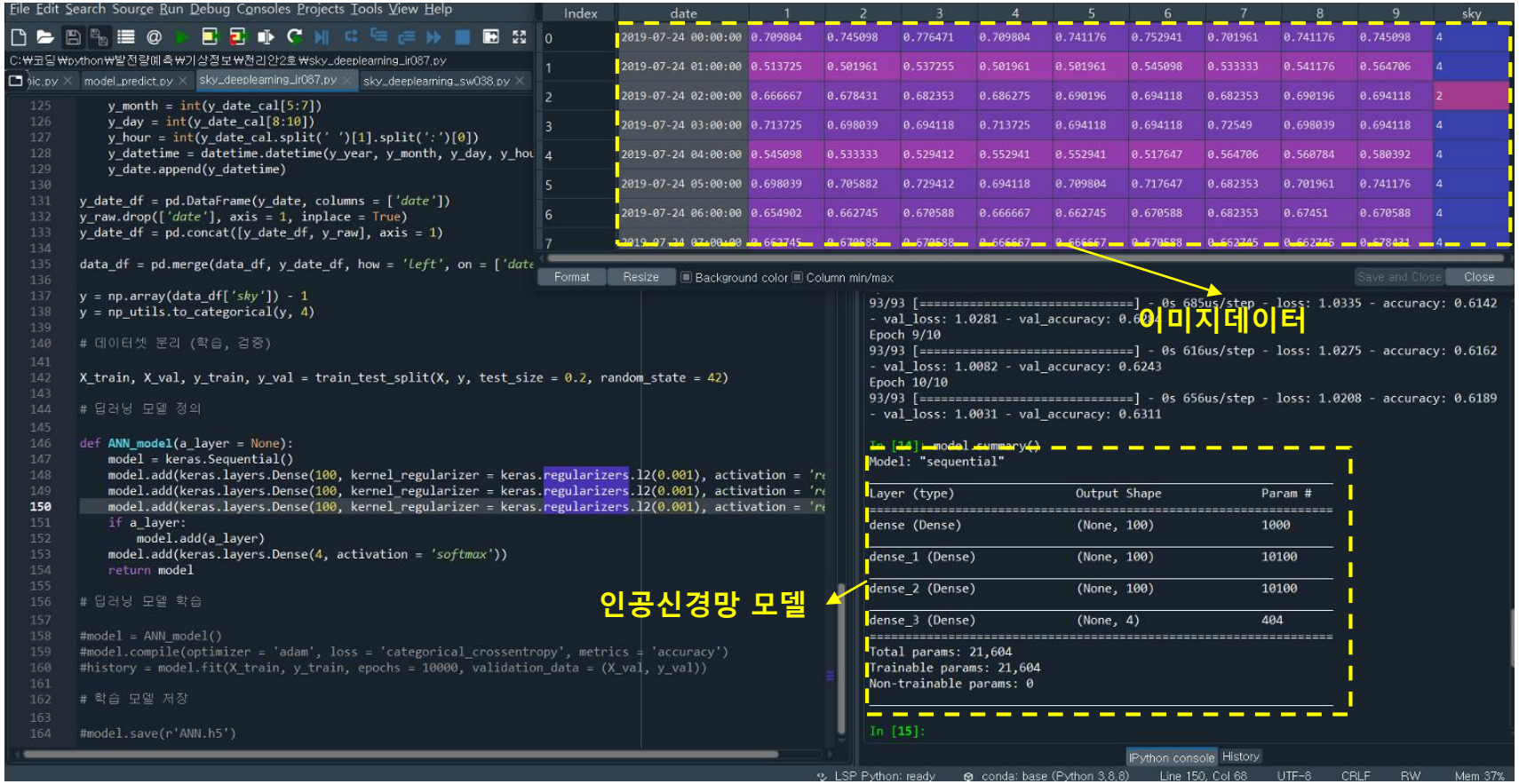
지역 이미지데이터 추출
(3×3)

이미지데이터 행렬 → 벡터
(3×3) → (9×1)

지역별 운량 산출

VI. 정확도 향상 방안 _ 향후 계획

- **천리안2호 위성 영상 데이터 샘플(6개월, '19. 7 ~ '19. 12)로 이미지데이터 추출 및 인공신경망 구현 완료**
- **현재 기상청 및 국가기상위성센터와 대용량 위성 영상 데이터 제공 가능 여부 협의 중**
  - 2년 2개월 ('19. 7 ~ '21. 8) 기간의 적외 8.7㎛, 주야간 합성 영상 총 120GB
- **데이터 취득 후 기 구현된 모델을 활용하여 머신러닝 수행 및 결과 분석 예정**



<이미지데이터 추출 및 인공신경망 구성>

첨 부

첨부. 재생E 보급에 따른 계통 불안정 (1/2)

- **태양광 등 출력이 수시로 증가하는 재생에너지의 증가로 전력계통 운영의 어려움 발생**
  - 예) 제주도 사례 : 낮시간 태양광 발전 증가 → 전력수요 초과 → 풍력 출력제한 발생 ('20. 7까지 13.7GWh)
- **전력계통의 안정성 확보를 위하여 재생에너지 발전량 예측과 제어를 위한 전력시장 개편 추진**

재생에너지 증가에 따른 계통 영향

재생에너지 증가 → **출력 변동성 증가** / **기존 발전기 대체** → **수급 유지의 곤란** / **시스템 운영 곤란** → **전력계통 안정성 저하 (주파수, 전압, 품질)**

전력시장 개편 추진

발전량 예측제도 ('20.9)	▶ **(대상) 20MW 이상의 태양광 및 풍력 발전사업자 및 소규모 전력중개사업자** ▶ **(조건) 하루전에 발전량 예측 제출, 예측오차율 8%이하인 경우, 3~4원/KWh의 정산금 지급** * '20.9. 시장운영규칙 개정 완료, '21 상반기 중 실행 예정
신재생발전량 입찰제도 (계획)	▶ **(개요) ESS, DR 등 활용 신재생발전기도 사전입찰한 발전량에 맞춰 발전, 급전지시, 출력제어** - 용량요금 지급 및 출력제어시 기회비용 보상 ▶ **(대상) 20MW 초과* 발전기 (단일규모 20MW 초과 및 중개사업자 모집자원 20MW 초과)**

첨부. 재생E 보급에 따른 계통 불안정 (1/2)

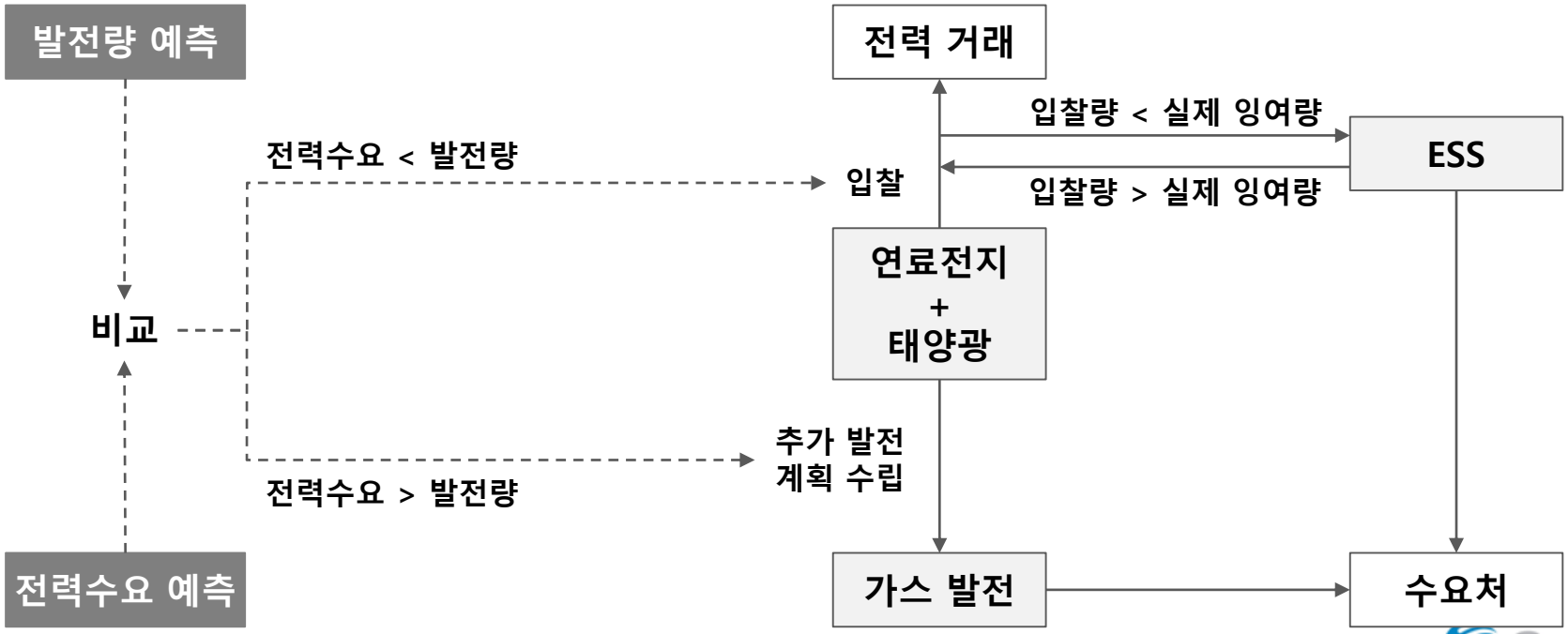
- '17년 IEA에서 변동성 재생E 보급 확대에 따른 전력계통 영향 및 대응 방안을 단계별로 제시
- '30년 이후 태양광 / 풍력 발전비율은 15%를 상회할 것으로 전망되며 이에 대응하기 위해 다양한 정책 제시
- 향후 재생E 입찰의무화, 분산E 활성화 등의 정책으로 현행 소규모 전력중개사업이 VPP로 확대 / 개편 가능성
- 소매의 경우 RE100 대응을 위한 재생E 제3자 PPA 제도 시행을 시작으로 재생전력판매사업 등으로 확대 검토

	전력계통 영향 ('17, IEA)	변동성 자원 보급률 (태양광/풍력, 9차)	제도 / 시장 변화	
			도매 시장	소매 시장
1단계	영향 無	'19년 2.6%	▪ 소규모 신재생E 전력중개사업	
2단계 (3% ~ 15%)	재생E 급전계획 반영 발전량 예측 도입	'20년 3.6%	▪ 전력중개사업의 VPP로의 확대 / 개편 - 1단계 : 재생에너지 발전량 예측 제도 → 예측 정확도에 따른 인센티브 지급 - 2단계 : 전력중개사업 대상 및 용량 확대 - 3단계 : 재생에너지 입찰 의무화 → 집합자원에 대한 용량 요금 지급 - 4단계 : 실시간/보조서비스 시장 개설 → 수익모델 다양화	▪ RE100 대응 - 제3자 PPA 시행 (재생E) ▪ 재생E P2P 검토
3단계 (15 ~ 25%)	유연성 자원 확대	'30년 14.6%		
4 ~ 6단계	P2X 일상화	'34년 19.4%	▪ 섹터 커플링의 활성화 및 소매시장 개방으로 종합에너지 공급 - Power2Heat, Power2Gas, Vehicle2Grid 등 * 섹터커플링 : 수소 등 에너지캐리어를 활용한 부문간 연계	

첨부. 지역에너지 자립 모델 _ 구역형 자립 모델

- 일정 구역 내에서 필요한 에너지를 그 구역 내에서 생산하여 충당하고 남는 에너지는 저장하거나 외부로 판매

구분	결합 모델	비고
발전량 > 수요량	전력 판매	▪ 실시간 시장 현황(가격, 수요 등)에 따른 우선 순위 산출 ▪ 우선순위에 따라 케이스 별로 스케줄 수립 및 운영 ▪ (예시) 구역형 에너지 자립 모델 - 발전량 예측 > 수요 예측 : 전력 시장에 판매 - 발전량 예측 < 수요 예측 : 추가 가스발전기 가동 - 실제 전력 거래 시 ESS 활용 입찰량 대비 판매량 정확도 제고
	ESS 저장	
	수소 생산	
	열 생산	
	전력 사용 장려	
발전량 < 수요량	추가 가스발전 가동	
	ESS 방전	
	외부 전력 구입	
	전력 사용 억제	



첨부. 업계 동향

- **재생E 발전량 예측 기술은 미래 가상발전소(VPP)의 핵심 기술로 평가**
- **재생E 발전량 예측 제도 실증 (’20. 11 ~ ’21. 6, KPX)의 태양광 분야에서 KT, SK E&S 등 총 10개사 합격**
  - 1개월간 태양광 발전 예측 오차율 평균 5~8% 달성 추정
  - 풍력 분야 합격사 없음
- **실증 합격사를 대상으로 제도 본격 시행 (’21. 10 예정) 시 등록 시험 면제**

구분		업체명	예측 기술		비고
			기술 보유 여부	협력사	
실증 참여 (합격)	1차	KT	O		오차율 8% 이하
		SK E&S	O		오차율 8% 이하
		솔라커넥트	O		오차율 8% 이하
	2차	한국남동발전	X	브이젠	기술 제공 비용 약 8천만원
		한국동서발전	X	인코어드	오차율 6% 이하
		해줌	O		서울대, 이화여대와 기술협력
		대건소프트	O		
		대연씨앤아이	O		
		안좌스마트팜앤쏠라시티	-		ESS 결합, 단일 발전사업자
		케이씨솔라앤에너지	-		
실증 미참여		씨엔씨티에너지	X	에코브레인	에코브레인의 2대주주
		포스코에너지	X	에코브레인	

첨부. 데이터 수집 (1/4) _ 태양 위치 계산

- 파이썬을 활용하여 태양광 발전소의 위도, 경도 등을 입력값으로 받아 태양의 고도, 방위각 산출
- 균시차 → 시간각 → 태양 적위 → 태양 고도 → 태양 방위각 순으로 계산 후 필요한 데이터 형태로 가공

1. 균시차	$ET(\text{균시차}) = 229.2 \times (0.000075 + 0.00186 \times \cos B - 0.032077 \times \sin B - 0.014615 \times \cos 2B - 0.04089 \times \sin 2B)$ $B = (n-1)\frac{360}{365}, \quad (n = Day\ of\ Year)$
2. 시간각	$\omega = \frac{LT \times 60 + 4 \times (LL - LST) + ET}{60} \times 15 - 180$ ω = 시간각, LT = 지역표준시, LL = 지역경도, LST = 표준 자오선 경도
3. 태양 적위	$\delta(\text{적위}) = 23.45 \times \sin\left[\frac{360}{365} \times (284 + Day\ of\ Year)\right]$
4. 태양 고도	$\sin\beta = \cos L \cos\delta \cos\omega + \sin L \sin\delta$ L = 지역경도, δ = 태양적위, ω = 시간각
5. 태양 방위각	$\cos\varphi = \frac{\sin\delta \cos L - \cos\omega \cos\delta \sin L}{\sin\theta}$ β = 태양고도, L = 지역경도, δ = 태양적위, θ = solar zenith angle = 90 - 태양고도

```
명령 프롬프트
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:WUsersWzepur>d:

D:W>cd 파이썬/발전량예측

D:W파이썬W발전량예측>python solar_altitude.py
              0   altitude  altitude_change     azimuth
0    2105090000 -35.170340        -3.168062  351.787418
1    2105090100 -35.054483         0.115857    9.317661
2    2105090200 -31.422800         3.631682   25.905397
3    2105090300 -24.872721         6.550079   40.373447
4    2105090400 -16.199619         8.673102   52.563579
5    2105090500  -6.093011        10.106608   62.921991
6    2105090600   4.941754        11.034766   72.066480
7    2105090700  16.545021        11.603266   80.628307
8    2105090800  28.437394        11.892373   89.290875
9    2105090900  40.344808        11.907414   98.980765
10   2105091000  51.875021        11.530213  111.336484
11   2105091100  62.195736        10.320715  129.847757
12   2105091200  69.124449         6.928713  160.871392
13   2105091300  68.860082        -0.264367  201.561731
14   2105091400  61.605791        -7.254291  231.627229
15   2105091500  51.164658       -10.441134  249.581122
16   2105091600  39.593168       -11.571490  261.691753
17   2105091700  27.677232       -11.915936  271.278832
18   2105091800  15.795484       -11.881749  279.911507
19   2105091900   4.219564       -11.575920  288.491218
20   2105092000  -6.768083       -10.987647  297.694162
21   2105092100 -16.800409       -10.032326  308.151169
22   2105092200 -25.360898        -8.560489  320.476526
23   2105092300 -31.748986        -6.388088  335.093605

D:W파이썬W발전량예측>
```

<결과 출력 화면>

첨부. 데이터 수집 (2/4) _ 어제 날씨 관측치 가져오기

- **공공데이터포털(www.data.go.kr)의 '기상청_지상(종관, ASOS) 시간자료 조회서비스' API 이용**
- **파이썬을 활용하여 오산, 광명과 근접한 수원 관측소 데이터를 가져옴**
  - 인증키, 일시, 관측소 번호 등을 파이썬 코딩을 통해 입력하여 데이터 요청
  - 획득한 데이터 중 발전량 예측에 필요한 데이터 (온도, 풍속, 습도, 운량 등)를 추출 및 정리

```
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:₩파이썬₩발전량예측₩whether_yesterday.py
predict_1(210504).py | predict_2(210506).py | predict_3(210506).py | predict_4(210507).py | predict_5(210507).py | forecastAPI_수치예보.py

# 필요 라이브러리 불러오기

import requests, json
import pandas as pd
import datetime

class whether:

    # 기상 데이터 불러오기 함수 get_forecast() 정의

    def get_whether(self, rows, start_date, start_hour, end_date, end_hour):
        self.servicekey = 'C5XWwKpv9QPJ8%2FX1nDI7qY2ZgVrDp9r4fzIXYCvjfj5uZ6MbPHY3SxTJkzrOpzDUrYW5j4%2FEJTk
        self.str = 'http://apis.data.go.kr/1360000/AsosHourlyInfoService/getWthrDataList?serviceKey=(servi
        self.url = self.str.format(servicekey=self.servicekey, page='1', rows=rows, data_type='json', star
        self.resp = requests.get(self.url)
        self.data = json.loads(self.resp.text)
        return self.data

    def make_whether_df(self, data):
        self.data_items = data['response']['body']['items']['item']

        self.df_data = pd.DataFrame(self.data_items)
        self.whether_df = self.df_data.drop(['stnId', 'rnum', 'taQcflg', 'rn', 'rnQcflg', 'wsQcflg', 'wd',
        self.whether_df.columns = ['date', 'location', 'temperature', 'windvelocity', 'humidity', 'sky']
        return self.whether_df

# 날짜 구하는 함수 yesterday(), today(), tomorrow() 정의

def yesterday():
    cdate = datetime.datetime.today()
    pdate = cdate - datetime.timedelta(days = 1)
    p_year = str(pdate.strftime("%y"))
    p_month = str(pdate.strftime("%m"))
    p_day = str(pdate.strftime("%d"))
    yesterday = '20' + p_year + p_month + p_day

    return yesterday
```

<데이터 요청 파이썬 코딩>

```
명령 프롬프트
D:₩>cd 파이썬/발전량예측

D:₩파이썬₩발전량예측>python whether_yesterday.py
                 date location temperature windvelocity humidity sky
0   2021-05-09 00:00       수원        10.6          0.6       93   0
1   2021-05-09 01:00       수원        10.7          1.2       91   0
2   2021-05-09 02:00       수원        10.5          1.4       92   0
3   2021-05-09 03:00       수원        10.2          1.0       96   0
4   2021-05-09 04:00       수원         9.3          0.2       96   0
5   2021-05-09 05:00       수원         8.9          1.2       97   0
6   2021-05-09 06:00       수원         8.0          0.7       98   0
7   2021-05-09 07:00       수원        11.7          0.3       91   0
8   2021-05-09 08:00       수원        12.8          2.8       80   0
9   2021-05-09 09:00       수원        15.4          4.2       63   0
10  2021-05-09 10:00       수원        17.3          5.0       52   0
11  2021-05-09 11:00       수원        19.2          5.9       40   0
12  2021-05-09 12:00       수원        19.7          6.3       34   0
13  2021-05-09 13:00       수원        20.3          5.4       32   0
14  2021-05-09 14:00       수원        20.1          5.9       33   0
15  2021-05-09 15:00       수원        20.4          6.0       28   0
16  2021-05-09 16:00       수원        20.8          5.0       29   0
17  2021-05-09 17:00       수원        19.9          3.6       34   0
18  2021-05-09 18:00       수원        17.8          4.9       38   4
19  2021-05-09 19:00       수원        16.4          3.8       42   0
20  2021-05-09 20:00       수원        14.3          1.6       56   0
21  2021-05-09 21:00       수원        12.8          1.4       64   0
22  2021-05-09 22:00       수원        12.0          0.6       71   0
23  2021-05-09 23:00       수원        11.0          1.2       77   0

D:₩파이썬₩발전량예측>
```

<결과 출력 화면>

첨부. 데이터 수집 (3/4) _ 내일 기상예보 가져오기

- **공공데이터포털(www.data.go.kr)의 '기상청_동네예보 조회서비스' API 이용**
- **파이썬을 활용하여 오산, 광명에서 가장 근접한 위치의 동네 예보 데이터를 가져옴**
  - 인증키, 일시, 사이트 위치 등을 파이썬 코딩을 통해 입력하여 데이터 요청
  - 획득한 데이터 중 발전량 예측에 필요한 데이터 (온도, 풍속, 습도, 운량 등)를 추출 및 정리

```
## 기상예보 클래스 forecast 정의

class forecast:

    # 기상 데이터 불러오기 함수 get_forecast() 정의

    def get_forecast(self, base_date, rows, base_time, nx, ny):
        self.servicekey = 'C5XWwKpv9QPJ8%2FX1nDI7qY2ZgVrDp9r4fzIXYCvjff5uZ6MbPHY3Sx'
        self.str = 'http://apis.data.go.kr/1360000/VilageFcstInfoService/getVilageF
        self.url = self.str.format(servicekey=self.servicekey, page='1', rows=rows,
        self.resp = requests.get(self.url)
        self.data = json.loads(self.resp.text)
        return self.data

    # 불러온 데이터 정리하기 함수 make_df() 정의

    def make_forecast_df(self, data):
        self.data_items = data['response']['body']['items']['item']

        self.df_data = pd.DataFrame(self.data_items)
        self.df_data_sort = self.df_data.sort_values(by = 'category')

        # 예보항목 = ['T3H(온도)', 'WSD(풍속)', 'REH(습도)', 'SKY(운량)']

        self.T3H = self.df_data_sort.loc[self.df_data_sort.category == 'T3H', :]
        self.T3H= self.T3H.sort_index()
        self.T3H.columns = ['announce_date', 'announce_time', 'announce', 'forecast_
        #self.T3H = self.T3H.set_index('forecast_time')
        self.T3H = self.T3H.drop(['announce', 'x', 'y'], axis = 1)

        self.WSD = self.df_data_sort.loc[self.df_data_sort.category == 'WSD', :]
        self.WSD = self.WSD.sort_index()
        self.WSD = self.WSD.drop(['baseDate', 'baseTime', 'category', 'nx', 'ny'],
        self.WSD.columns = ['forecast_date', 'forecast_time', 'windvelocity']
        #self.WSD = self.WSD.set_index('forecast_time')
        self.forecast = pd.merge(self.T3H, self.WSD, how = 'left', on = ['forecast_

        self.REH = self.df_data_sort.loc[self.df_data_sort.category == 'REH', :]
        self.REH = self.REH.sort_index()
        self.REH = self.REH.drop(['baseDate', 'baseTime', 'category', 'nx', 'ny'],
        self.REH.columns = ['forecast_date', 'forecast_time', 'humidity']
        #self.REH = self.REH.set_index('forecast_time')
        self.forecast = pd.merge(self.forecast, self.REH, how = 'left', on = ['fore

        self.SKY = self.df_data_sort.loc[self.df_data_sort.category == 'SKY', :]
```

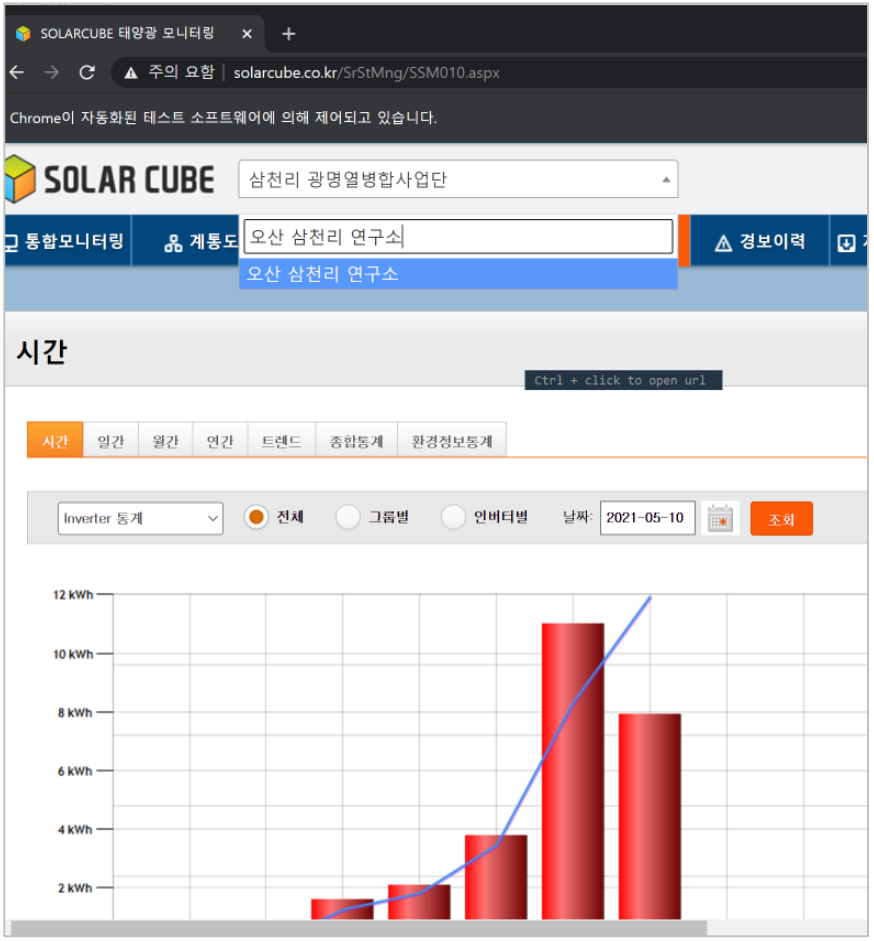
<데이터 요청 파이썬 코딩>

```
명령 프롬프트
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\zepur>d:

D:\>cd 파이썬/발전량예측

D:\파이썬\발전량예측>python forecastAPI(210414)_class.py
          announce_date announce_time forecast_date temperature windvelocity humidity sky
forecast_time
0000           20210510          0800      20210511          12         1.2       90   4
0300           20210510          0800      20210511          11         1.8       90   3
0600           20210510          0800      20210511          10         1.9       85   1
0900           20210510          0800      20210511          16         3.1       65   1
1200           20210510          0800      20210511          21         4.3       40   1
1500           20210510          0800      20210511          23         4.9       35   1
1800           20210510          0800      20210511          21         2.9       40   1
2100           20210510          0800      20210511          17         3.6       45   1
          announce_date announce_time forecast_date temperature windvelocity humidity sky
forecast_time
0000           20210510          1100      20210511          12         1.2       90   4
0300           20210510          1100      20210511          11         1.8       90   3
0600           20210510          1100      20210511          10         1.9       85   1
0900           20210510          1100      20210511          16         3.1       65   1
1200           20210510          1100      20210511          21         4.3       40   1
1500           20210510          1100      20210511          23         4.9       35   1
1800           20210510          1100      20210511          21         2.9       40   1
2100           20210510          1100      20210511          17         3.6       45   1

D:\파이썬\발전량예측>
```

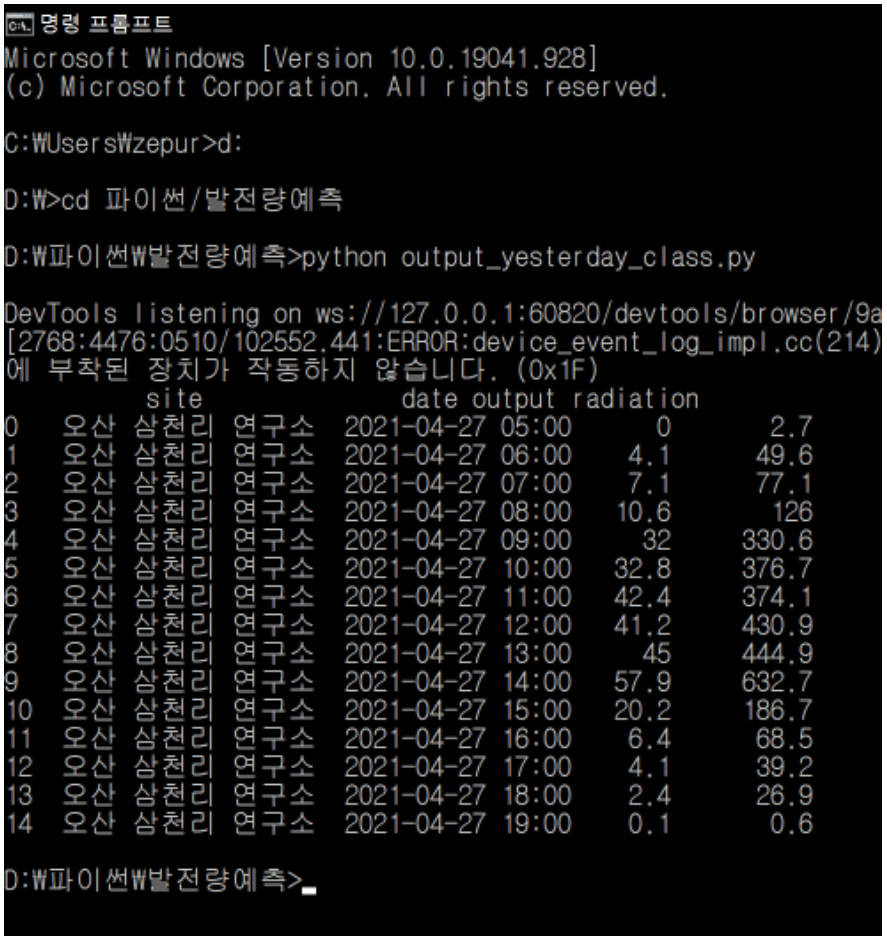
<결과 출력 화면>

첨부. 데이터 수집 (4/4) _ 발전량 데이터 가져오기

- **파이썬 코딩을 통해 크롬 브라우저를 자동 제어하여 태양광 모니터링 사이트에서 데이터를 추출**
  - 사이트 접속 → ID/PASSWORD 입력 → 메뉴 선택 → 사이트/날짜 입력 → html 소스 불러오기 → 데이터 추출



<크롬브라우저 자동 제어>



<데이터 추출 화면>

첨부. 예측 예시

시간	태양 위치 데이터			기상 데이터					발전량
	고도	고도변화	방위각	기온	강수량	풍속	습도	운량	
10	19.38	8.20	142.27	-3.2	0	0.7	55	3	70.7
				.....					.....
17	10.96	-10.92	251.82	18.8	0	1.7	45	2	8.1

<학습데이터>

① 학습모델 선택 및 데이터 학습

예측 모델

③ 예측

발전량 예측
138.7

② 예측 모델에 예측 기초데이터 입력

시간	태양 위치 데이터			기상 데이터				
	고도	고도변화	방위각	기온	강수량	풍속	습도	운량
12	45.00	3.46	173.15	17.9	0	2.8	34	1

<예측 기초 데이터>

첨부. 학습모델 세부 내용 (1/4) _ 선형회귀

- 선형회귀(linear regression)는 종속변수와 한 개 이상의 독립변수의 선형상관관계를 모델링하는 회귀분석 기법
- 머신러닝 분야에서는 평균제곱오차(mean square error)를 손실함수로 한 경사하강법을 사용하여 모델 구축

경사하강법은 손실함수인 평균제곱오차 그래프의 기울기가 최소인 지점에서 오차도 최소가 된다는 사실을 이용

$$\text{손실함수} = \text{평균제곱오차} = \frac{1}{n}\sum_{i}^{n}(\hat{y}_i - y_i)^2$$

손실함수의 기울기를 구하기 위해서는 손실함수를 미분하고 그 값이 0이 되는 지점이 오차가 최소인 지점

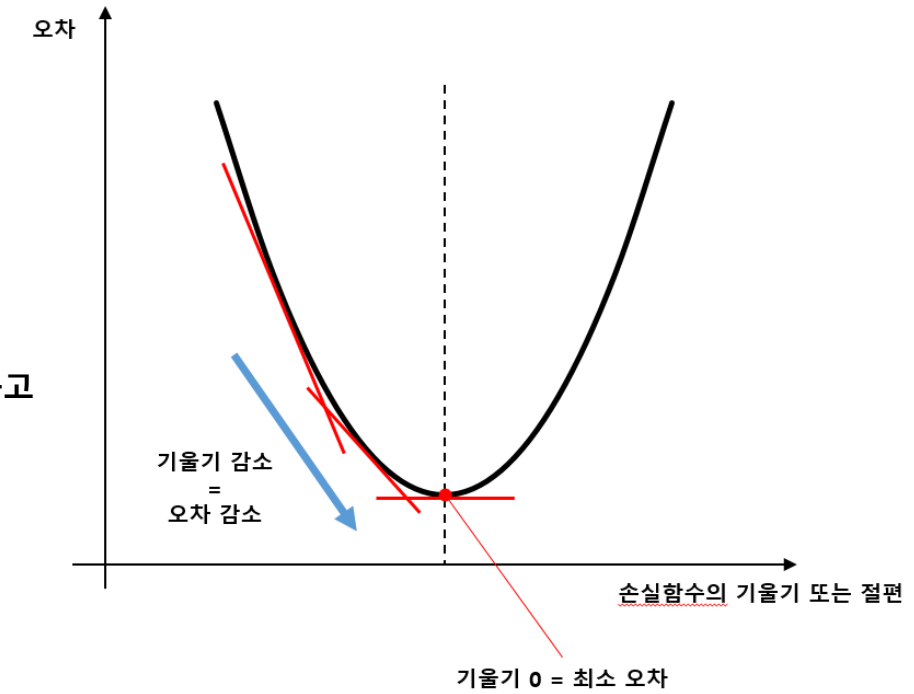
손실함수를 미분하기 위해 선형회귀모델을 정의하고 손실함수에 대입하고 편미분하면 다음과 같다.

$$\hat{y} = ax_i + b$$

$$L(\text{손실함수}) = \frac{1}{n}\sum_{i}^{n}(ax_i + b - y_i)^2$$

$$\frac{\partial L}{\partial a} = \frac{2}{n}\sum_{i}^{n}(ax_i + b - y_i)x_i$$

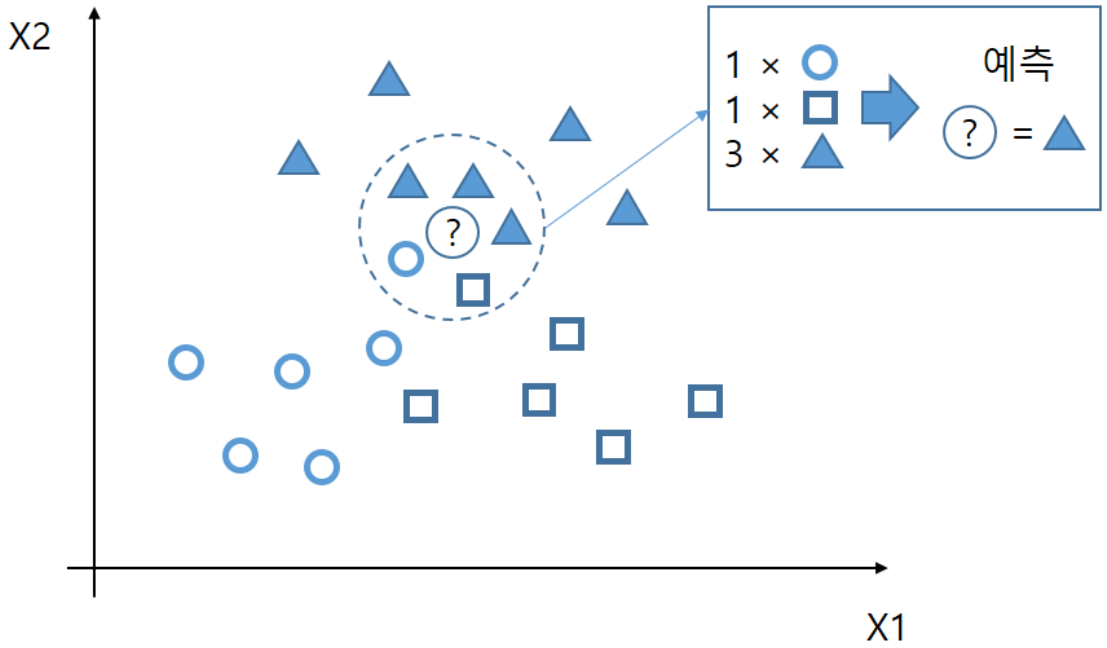
a값을 점진적으로 증가 또는 감소시키면서 대입하는 것을 반복하면 기울기가 0이 되는 지점을 찾아 선형회귀모델을 최적화



<경사하강법>

첨부. 학습모델 세부 내용 (2/4) _ k-최근접이웃(kNN)

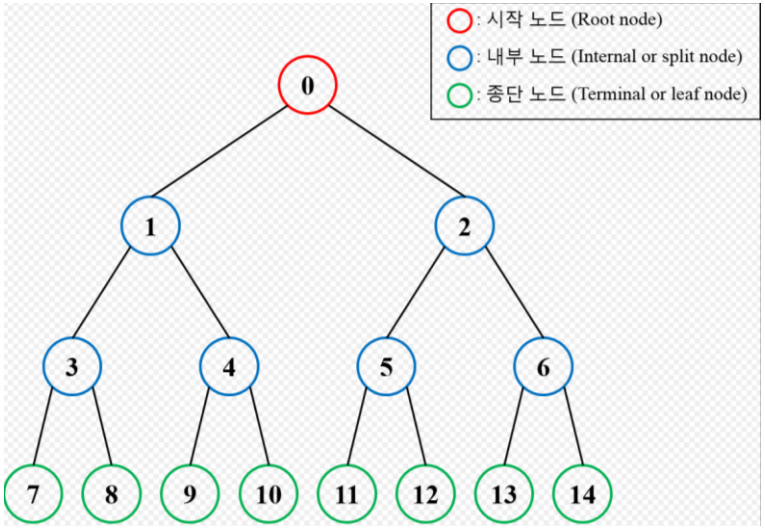
- **K-최근접이웃 (k-nearest neighbor)은 다른 알고리즘과 비교해 간단한 편으로 다음 단계로 요약 가능**
  - 숫자 k와 거리 측정 기준 선택
  - 분류하는 샘플에서 k개의 최근접 이웃을 찾음
  - 다수결 투표를 통해 클래스 레이블 할당
- **거리 측정방식은 유클리디안(Euclidean) 거리 측정 방식 사용**
  - **두 점을 연결한 유클리디안 벡터의 놈(norm)을 구함** $E = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



<kNN 알고리즘의 다수결 투표>

첨부. 학습모델 세부 내용 (3/4) _ 랜덤포레스트

- **랜덤포레스트(random forest)는 다수의 결정트리들을 학습하는 앙상블 기법으로 다양한 문제해결에 활용**
- **단순한 결정트리의 결과를 일반화하기 어렵다는 단점 해결 가능**
- **기본적인 아이디어는 여러 개의 결정트리의 평균을 구하는 것으로 다음 네 단계로 요약 가능**
  - 1단계 : 훈련데이터에서 랜덤하게 n개의 샘플 선택 (중복 허용)
  - 2단계 : 선택한 샘플에서 결정트리 학습
  - 3단계 : 단계 1~2를 k번 반복
  - 4단계 : 각 트리의 예측을 모아 다수결 투표로 결정



<결정트리 예시>

결정트리는 결정을 내리기 위해 사용하는 트리로 결정과정을 간단한 문제들로 이루어진 계층 구조로 이루어지며 노드(node)와 에지(edge)의 집합으로 구성
노드는 내부노드와 종단노드로 나뉘며 모든 노드는 들어오는 에지는 유일
반면 각 내부노드에서 나가는 에지의 개수는 제한없음

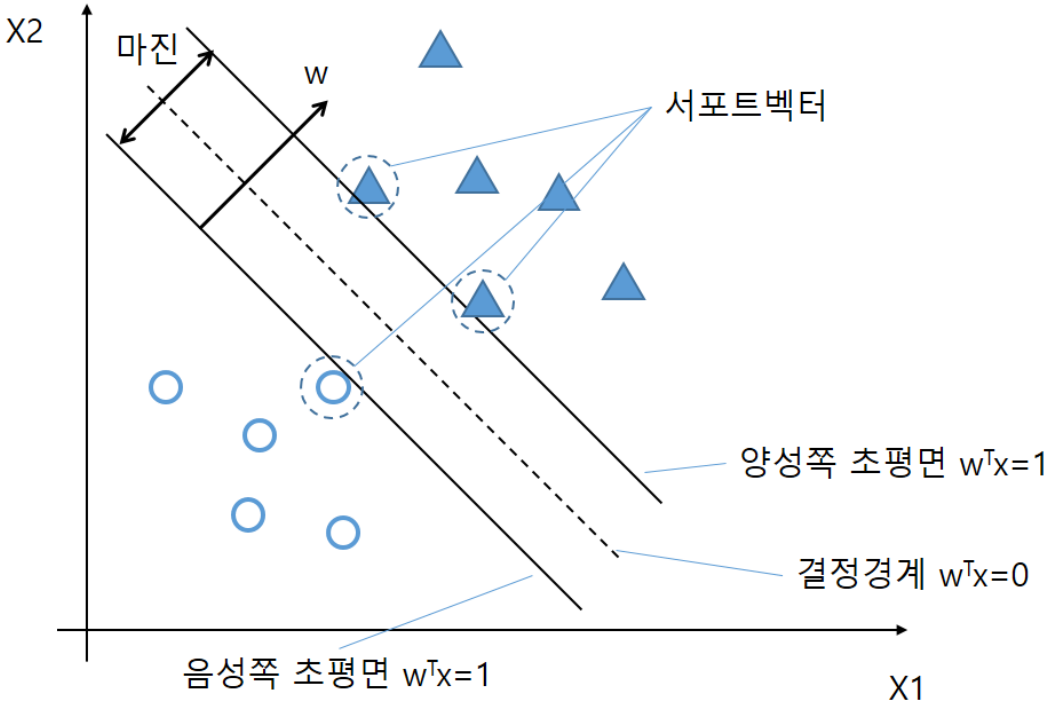
결정트리 알고리즘은 트리의 시작 노드부터 정보이득이 최대가 되는 특성으로 데이터를 나누며 정보이득(IG)는 다음과 같다.

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

※ 자세한 내용은 '에너지 거래 기술라이브러리' 문서 참조

첨부. 학습모델 세부 내용 (4/4) _ 서포트벡터머신(SVM)

- **서포트벡터머신 (support vector machine, SVM)은 강력한 성능으로 가장 널리 사용되는 학습 알고리즘**
- **클래스를 구분하는 결정 경계의 마진 (결정 경계와 최근접 훈련 샘플과의 거리)을 최대화하여 최적화**
  - 마진을 최대화하는 이유는 일반화가 용이, 반대의 경우 학습 정확도는 높아지지만 예측 정확도는 낮아짐



<서포트벡터머신>

양성쪽과 음성쪽의 초평면은 다음과 같이 표현 가능

$$w_0 + \boldsymbol{w}^T \boldsymbol{x}_+ = 1 \quad \textbf{(1)}$$

$$w_0 + \boldsymbol{w}^T \boldsymbol{x}_- = -1 \quad \textbf{(2)}$$

두 선형식 (1)과 (2)를 빼면 다음과 같으며 벡터 w의 길이로 정규화 가능

$$\boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-) = 2$$

$$\| \boldsymbol{w} \| = \sqrt{\sum_{j=1}^{m} w_j^2}$$

$$\frac{\boldsymbol{w}^T(\boldsymbol{x}_+ - \boldsymbol{x}_-)}{\| \boldsymbol{w} \|} = \frac{2}{\| \boldsymbol{w} \|}$$

위 식의 좌변은 양성쪽 초평면과 음성쪽 초평면의 거리로 해석가능하며 이것이 최대화하려는 마진

첨부. 세부 예측 결과 (1/2)

▪ 기간1

구분		다중 선형회귀	랜덤포레스트	KNN	서포트벡터머신
오차율	오산	12.11%	11.90%	10.43%	10.95%
	광명	11.92%	12.72%	11.55%	11.53%
	합계	**10.29%**	**11.21%**	**9.98%**	**10.70%**
인센티브	오산	25,843	27,789	25,526	27,527
	광명	36,756	34,044	30,169	34,628
	합계	**70,505**	**65,141**	**67,484**	**60,841**
최대 인센티브	오산	61,682	61,682	61,682	61,682
	광명	74,416	74,416	74,416	74,416
	합계	**135,954**	**135,954**	**135,954**	**135,954**
인센티브 획득률	오산	41.9%	45.1%	41.4%	44.6%
	광명	49.4%	45.7%	40.5%	46.5%
	합계	**51.9%**	**47.9%**	**49.6%**	**44.8%**

▪ 기간2 (함평 제외)

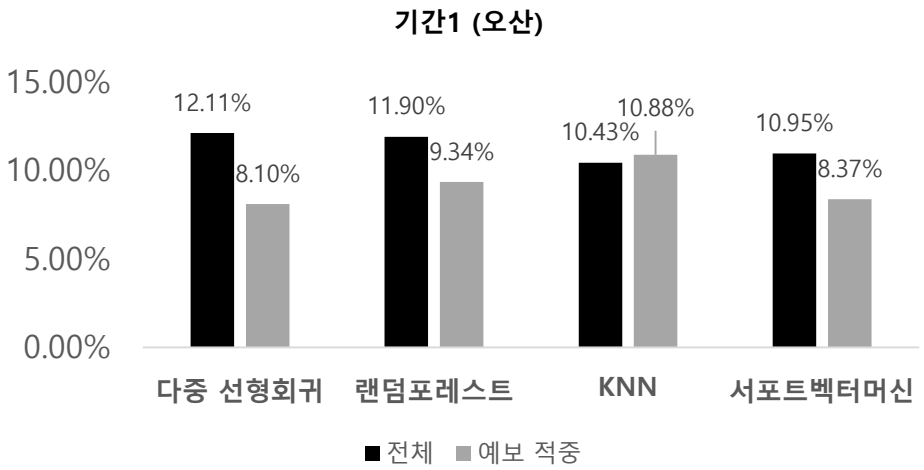
구분		다중 선형회귀	랜덤포레스트	KNN	서포트벡터머신
오차율	오산	9.21%	9.43%	8.32%	9.19%
	광명	10.33%	11.67%	8.86%	10.12%
	합계	**8.82%**	**9.38%**	**7.42%**	**8.65%**
인센티브	오산	37,246	44,017	40,871	41,391
	광명	42,423	44,456	49,401	47,971
	합계	**90,283**	**93,184**	**100,602**	**96,915**
최대 인센티브	오산	80,020	80,020	80,020	80,020
	광명	95,508	95,508	95,508	95,508
	합계	**175,224**	**175,224**	**175,224**	**175,224**
인센티브 획득률	오산	46.5%	55.0%	51.1%	51.7%
	광명	44.4%	46.5%	51.7%	50.2%
	합계	**51.5%**	**53.2%**	**57.4%**	**55.3%**

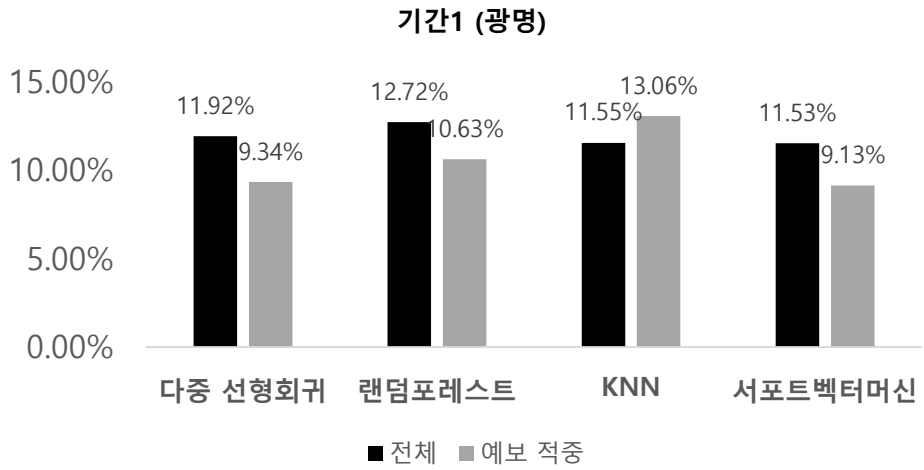
▪ 기간2 (함평 포함, 서포트벡터머신)

구분	오차율	획득 인센티브	최대 가능 인센티브	인센티브 획득률
오산	9.19%	41,391	80,020	51.7%
광명	10.12%	47,971	95,508	50.2%
함평	9.96%	66,597	147,790	45.1%
합계	**7.20%**	**182,279**	**316,654**	**57.6%**

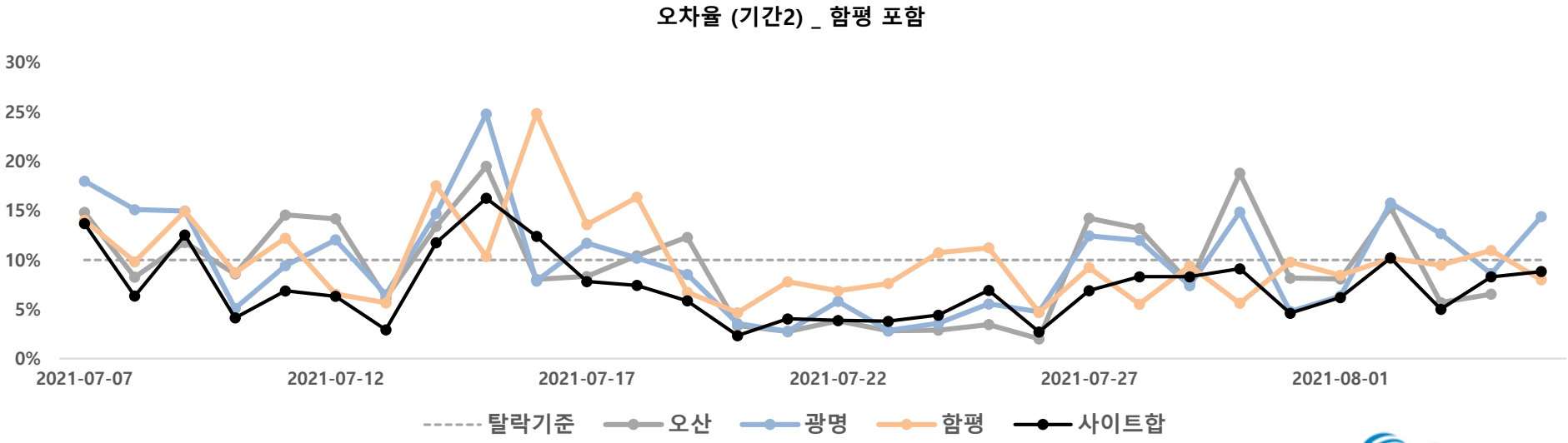
첨부. 세부 예측 결과 (2/2)

▪ 기상예보 적중 유무에 따른 오차율 변화



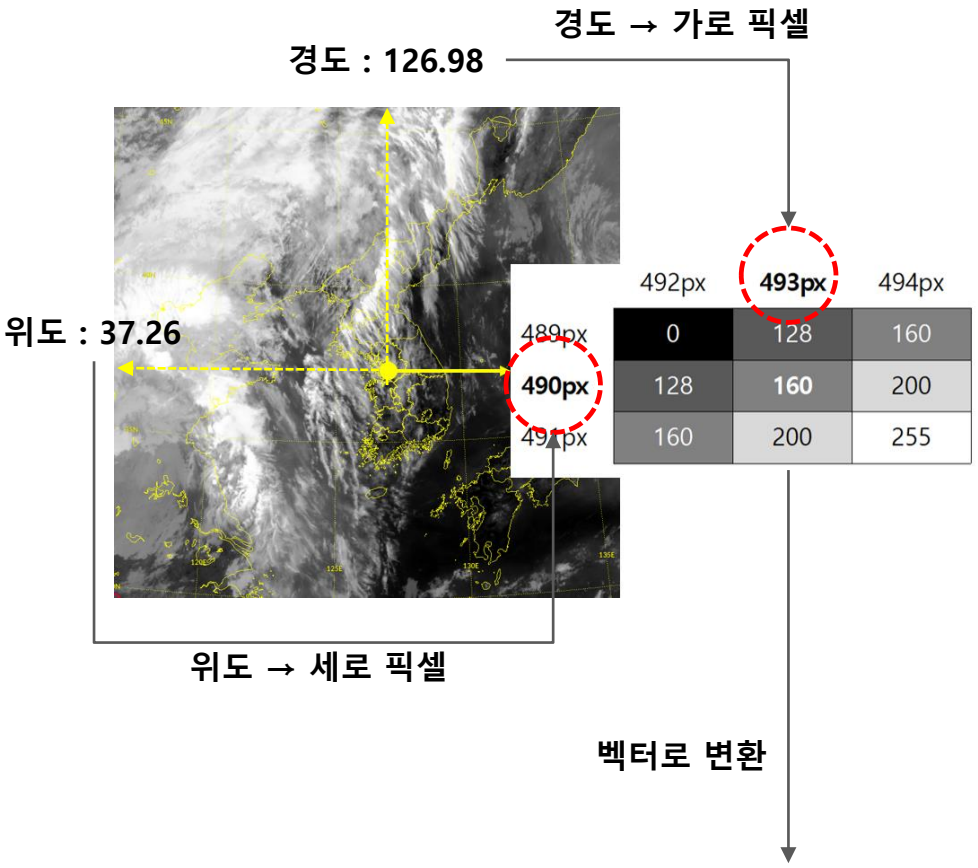


▪ 일자별 세부 오차율



첨부. 이미지 처리 및 예측 예시

▪ 수원 기상관측소 상공 이미지 추출 예시



[0, 128, 160, 128, 160, 200, 160, 200, 255]

이미지데이터 추출 및 벡터 변환

이미지 데이터 (오산 기술연구소 상공)								
(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
0	50	65	100	120	115	50	60	100

오산기술연구소 상공
이미지데이터 입력

예측

예측 모델

운량 예측
4

인공신경망

이미지 데이터 (수원 기상관측소 상공)									운량
(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	
0	128	160	128	160	200	160	200	255	3

첨부. 인공신경망

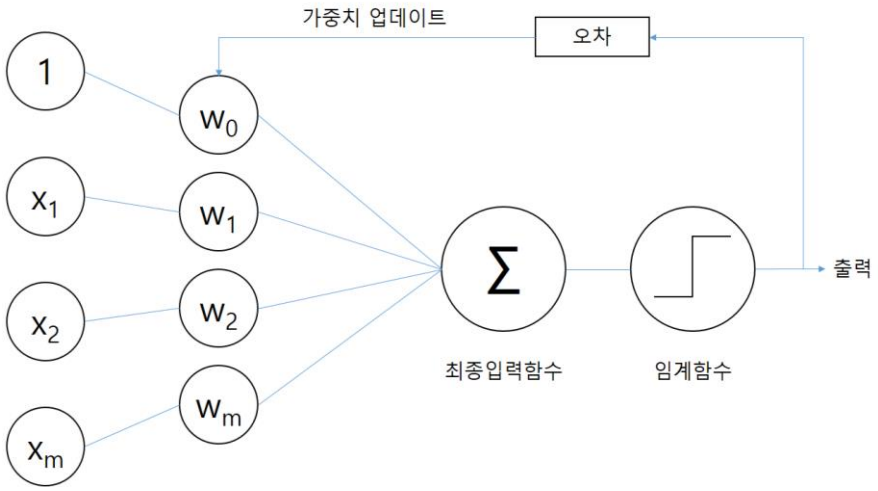
- 인공신경망(딥러닝)은 머신러닝의 하위분야로 생물학적 뇌의 신경세포에 도달한 신호가 특정 임계값을 넘으면 출력 신호가 생성되어 다음 신경세포로 전달되는 원리를 이용
- 최적의 가중치를 자동으로 학습하는 퍼셉트론 개념으로 새로운 데이터가 어떤 클래스에 속하는지 예측 가능

인공뉴런은 이진분류작업이며 두 클래스를 1과 -1로 나타낼 수 있다. 입력값 x에 상응하는 가중치 벡터 w의 선형함수로 결정함수 정의

최종입력 z는 $z = w_1x_1 + w_2x_2 + \cdots + w_mx_m$

입력치 x의 최종 입력값이 사전에 정의된 임계값 θ보다 크면 1, 그렇지 않으면 -1로 예측하고 이를 결정함수로 표현하면 다음과 같다.

$$\Phi(z) = \begin{cases} 1 & z \geq \theta \text{ 일 때} \\ -1 & \text{그외} \end{cases}$$



<퍼셉트론 알고리즘>

임계값 θ를 좌변으로 옮겨 w_0=-θ이고 x_0=1인 0번째 가중치를 정의

z를 간단하게 정리하고 이에 따른 결정함수는 다음과 같다.

$$z = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \boldsymbol{w}^T\boldsymbol{x}$$

$$\Phi(z) = \begin{cases} 1 & z \geq 0 \text{ 일 때} \\ -1 & \text{그외} \end{cases}$$

퍼셉트론 알고리즘에 따른 가중치 업데이트 수식과 업데이트값 Δw_j

$$w_j \coloneqq w_j + \Delta w_j$$

$$\Delta w_j = \eta(y^{(i)} - \hat{y}^{(i)})x_j^{(i)}$$

여기서 η는 학습률 ^가 있는 y는 예측값이다.

※ 자세한 내용은 '에너지 거래 기술라이브러리' 문서 참조