기계학습을 적용한 위성관측 데이터 활용 강수량 산출

2020년 1학기

기계학습 및 실습 레포트

생명과학과 2015560004 김보경

전자전기컴퓨터공학부 2017440064 부대권

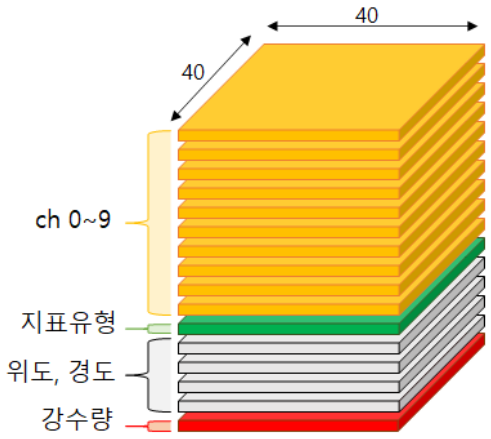
I. 서론

[요약]

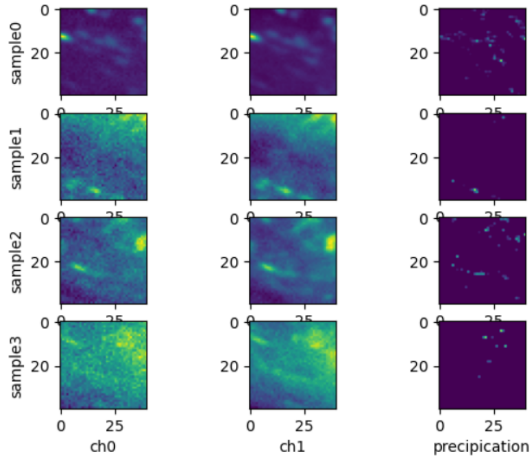
데이터 사이언스 관련 대회를 주최하는 DACON에서 “위성관측 데이터 활용 강수량 산출 AI 경진대회”를 개최하였다. 이 대회는 인공위성(GMI센서)으로 한국 주변지역을 촬영한 이미지와 지표유형(바다, 연안, 섬, 육지, 호수)을 설명변수로 사용하여 “강수량”이라는 반응변수를 예측하는 것이 목적이다. 이 대회에서 제공하는 데이터의 일부를 사용하여 기계학습 수업에서 배운 내용을 적용한 기계학습 모델을 만들고자 한다.

[데이터에 대한 설명]

대회에서 제공하는 데이터는 이미지는 76,345개 이며, 하나의 이미지 샘플은 40*40 (=1600)개의 픽셀을 가진다. 따라서 총 데이터의 수는 122,152,000개이다. 한 픽셀 마다 14개의 설명변수와 1개의 반응변수로 이루어진다.



[그림1.] 하나의 샘플이 갖는 설명변수와 반응변수



[그림2.] 4개의 샘플에서 0번,1번 채널과 강수량(precipitation)을 이미지로 표현

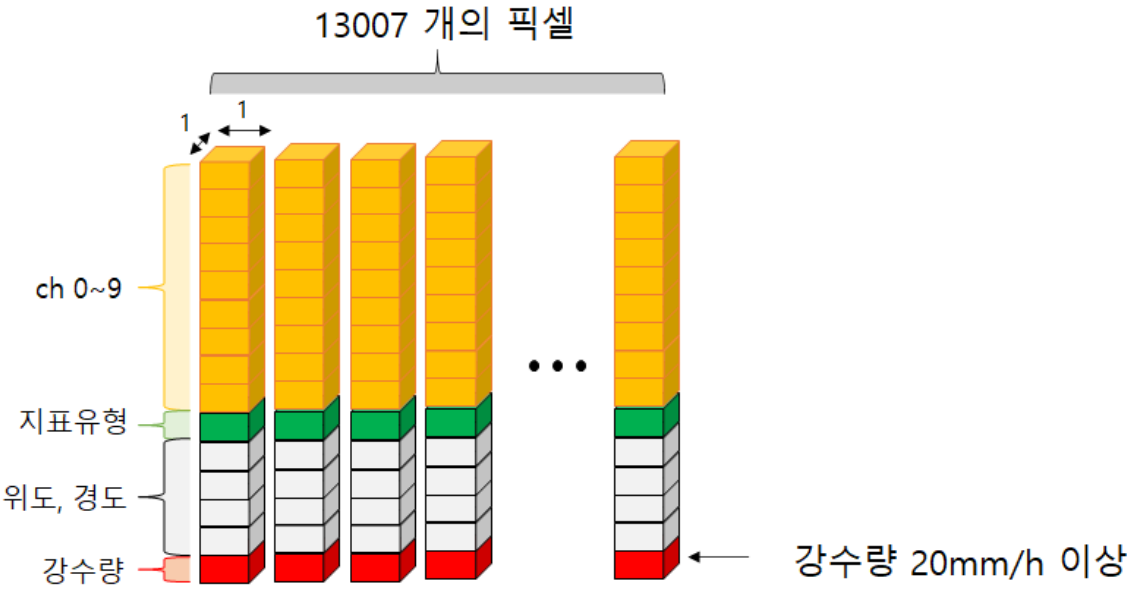
설명변수	
ch 0 ~ 8	GPM(Global Precipitation Measurement) Core 위성의 9가지 GMI센서로 촬영한 이미지, 밝기 온도 (단위: K, 10.65GHz~89.0GHz)
지표유형	앞자리 0: Ocean, 앞자리 1: Land, 앞자리 2: Coastal, 앞자리 3: Inland Water
위도, 경도	GMI센서의 위도, GMI센서의 경도, DPR센서의 위도, DPR센서의 경도 두 센서 모두 위도(북위 약 5 º ~ 55 º), 경도(약 105 º ~ 175 º) 의 영역에 분포한다.

반응변수	
강수량	GPM core 위성의 DPR 센서로 측정한 강수량 (mm/h, 결측치는 -9999.xxx 형태의 float 값으로 표기)

[분석 목표]

기말 프로젝트 레포트는 수업에서 배운 내용을 토대로 논리적으로 문제에 접근하여 결론을 내리는 것을 목표로 한다. 이 대회에서 제공하는 모든 데이터를 이용하여 레포트를 작성하는 데는 사용 가능한 자원이 부족하다고 판단하여 일부 데이터만을 사용하기로 결정하였다.

따라서 이 대회에서 제공하는 데이터 중 랜덤으로 10,000개의 이미지를 추출하고, 그 중에서 강수량이 20이 넘는 데이터만을 사용한다. 가공한 데이터는 [그림3.] 과 같다. 수업 내용을 활용하여 위의 주어진 설명변수로 반응 변수를 가장 잘 예측(회귀값)하는 모델을 만드는 것을 목표로 한다.



[그림3.] 학습에 사용한 데이터

EDA를 통해서 Outlier(이상치), High Leverage Point(영향 관측값), 결측값 등을 찾아 처리하고, 여러가지 기계학습 기법(다항회귀, SVM, Tree based model, 표본재추출)을 적용한다. 또한 PCA(주성분 분석)방법을 적용한 데이터로 모델을 학습하고 기존의 방법과 성능을 비교한다.

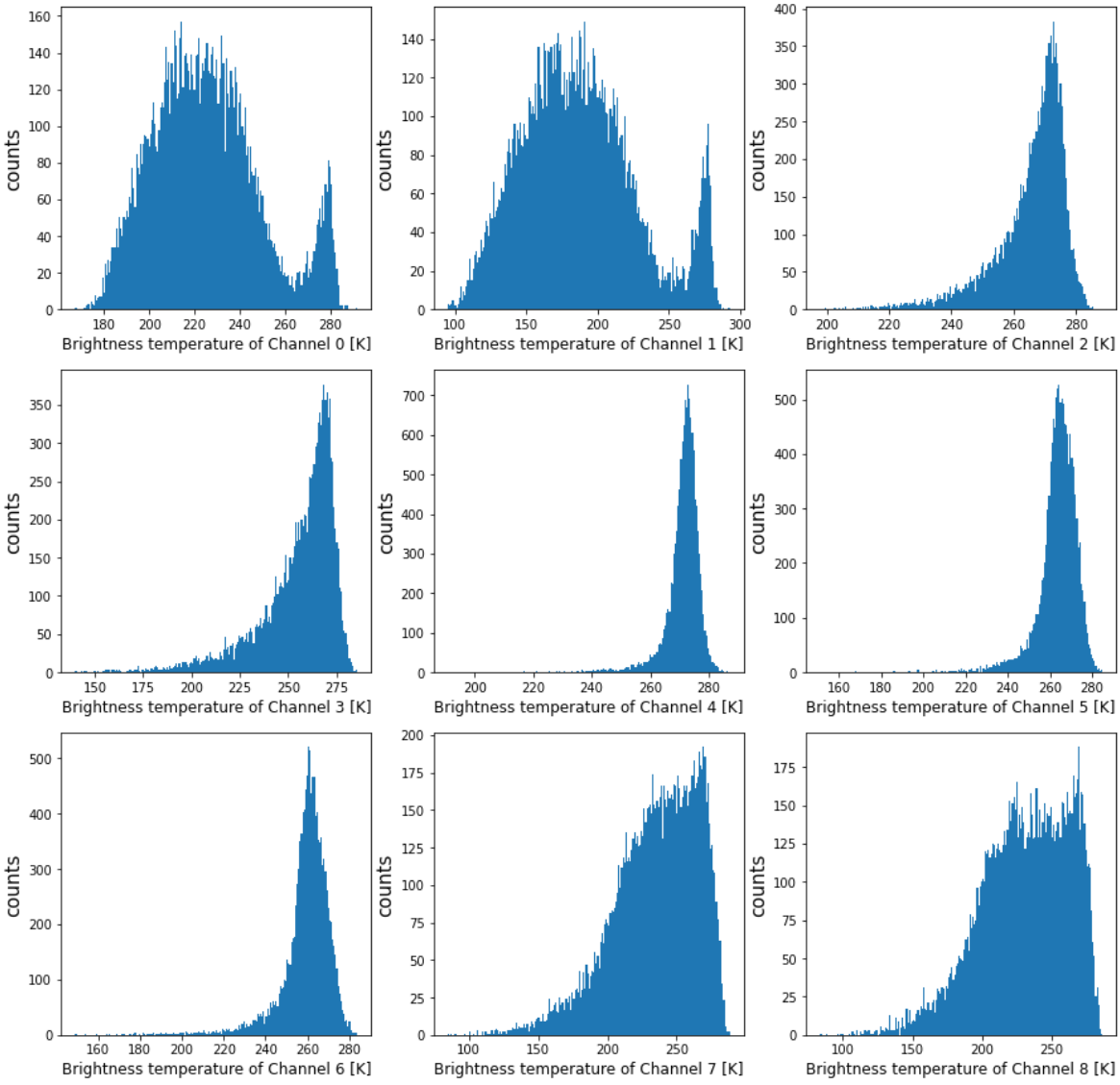
II. 본론

A. 데이터 EDA(Exploratory data analysis)

1) 설명변수 EDA 및 전처리

a) 센서 데이터(9개)의 분포 시각화

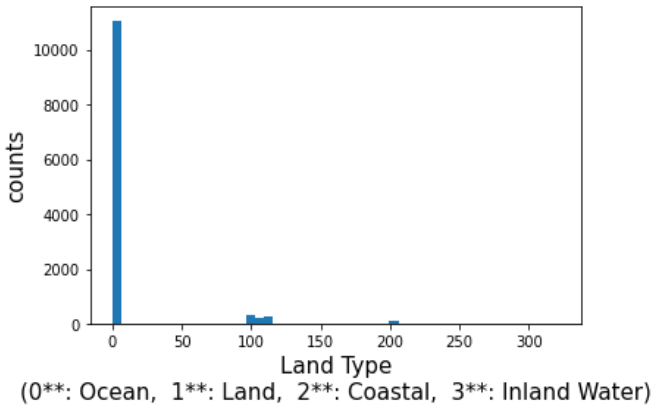
Channel의 밝기 값들은 100[K]~ 300[K] 사이에 분포한다. 0번과 1번 채널에서는 2개의 극댓값이 존재하며, 2번 채널부터는 전체적인 분포가 우로 편향된 모습을 보인다.



[그림 4.] 9개 GPM 센서 밝기 온도값의 분포

b) 지표유형 변수의 분포 시각화

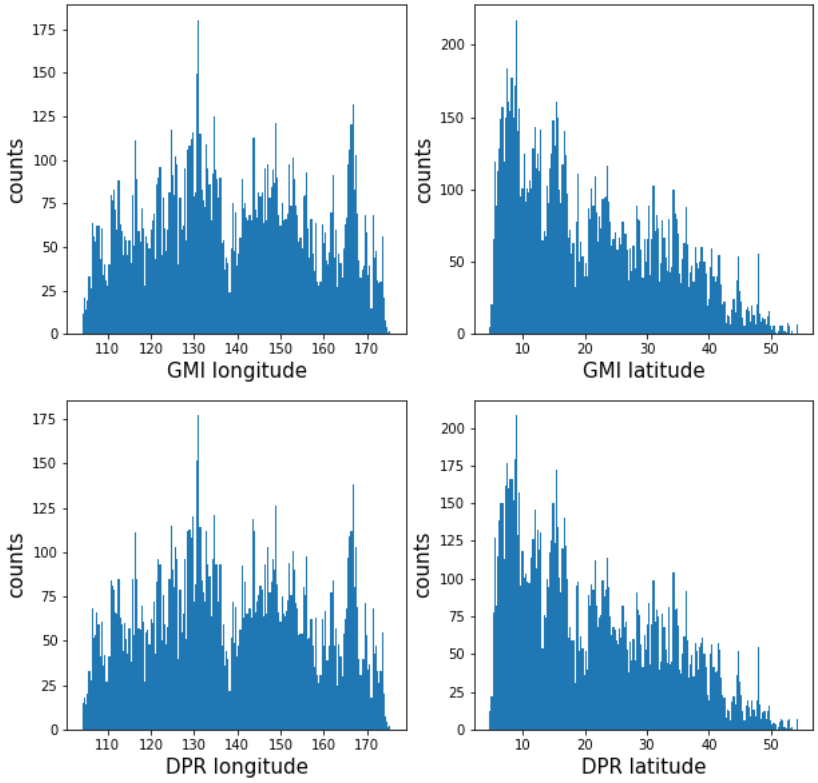
대부분의 데이터가 바다에 존재한다.



[그림 5.] 지표유형 변수의 분포

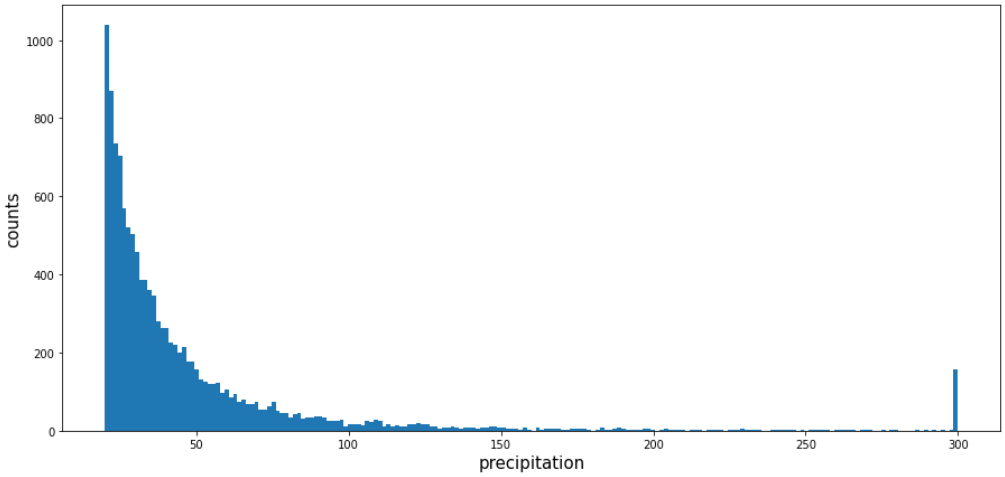
c) 위도, 경도 변수의 분포 시각화

GMI센서와 DPR센서의 위도 경도값에는 큰 차이가 없다. 경도는 불규칙하게 분포하며, 위도는 저위도의 데이터가 많이 존재한다.



[그림 6.] 위도, 경도 변수의 분포

d) 반응 변수(강수량)의 분포 시각화



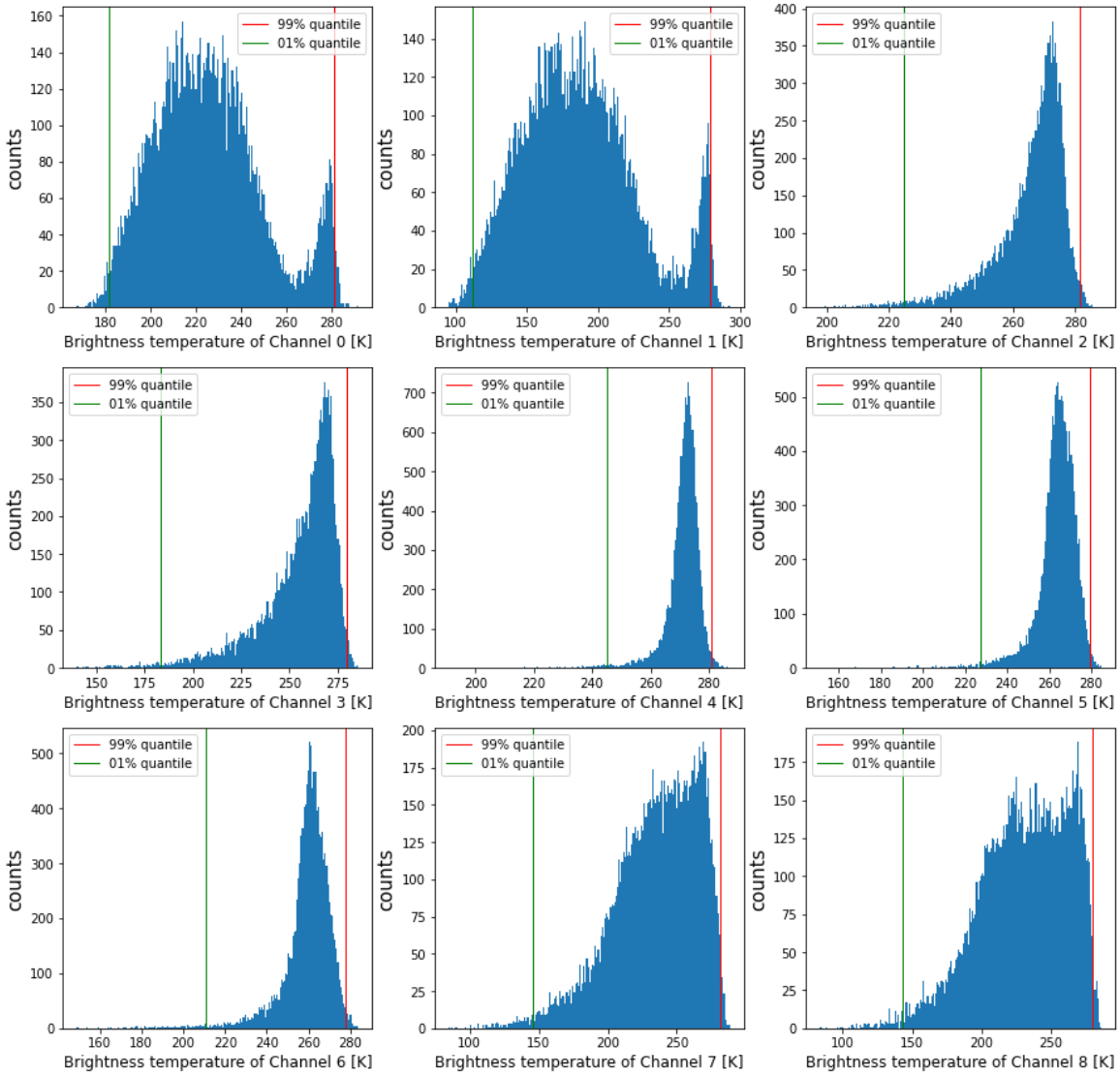
[그림 7.] 반응변수(강수량)의 분포

반응변수의 평균, 표준편차는 다음과 같다.

평균	41.649
표준 편차	25.158

전체 데이터는 0mm/h 와 300mm/h사이의 값을 갖는다. 0mm/h 에서 가장 많이 존재하며, 수가 지수적으로 감소한다. 특이하게 299mm/h ~ 300mm/h구간에서 많은 값이 분포한다.

e) High leverage point(영향 관측값) 탐색



[그림 8.] 9개 GPM 센서 밝기 온도값의 분포에서 영향 관측값 탐색

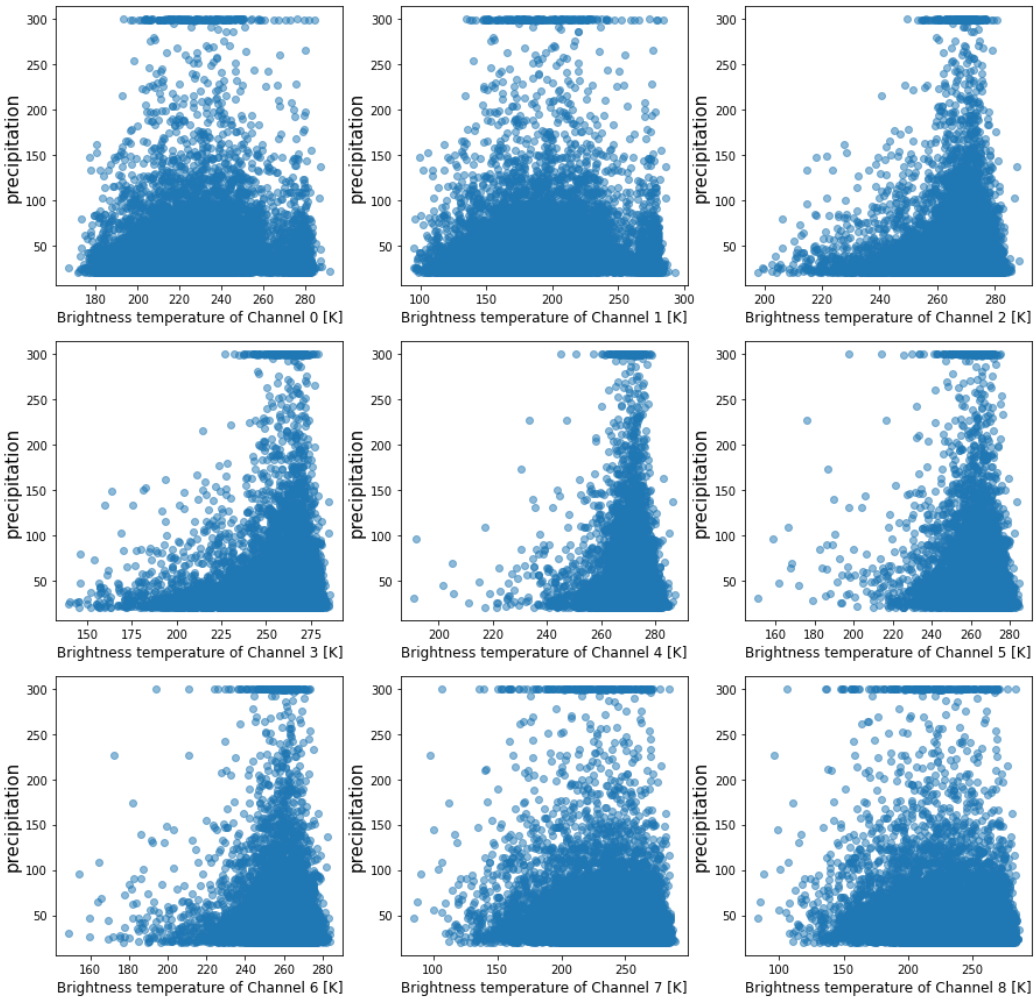
설명변수의 값이 다른 데이터들과 동떨어진 것들을 영향 관측값이라고 한다. 설명변수의 분포를 시각화 했을 때 일부 채널에서 매우적은 데이터가 상대적으로 넓은 범위에 분포하는 모습을 보였다. 따라서 각 채널별로 1% Quantile 이하, 99% Quantile 이상의 데이터를 영향 관측 값이라고 설정했다.

Land Type 데이터는 대부분의 데이터가 Ocean에 몰려있긴 하지만 4가지 종류로 분류된 변수이기 때문에 영향 관측값을 따로 지정하지 않았다.

위도, 경도 데이터는 비교적 일정하게 분포되어 있어 영향 관측값이 나타나지 않았다.

2) 설명변수와 반응변수의 관계 조사

a) 설명변수와 반응변수의 산점도(scatter plot)

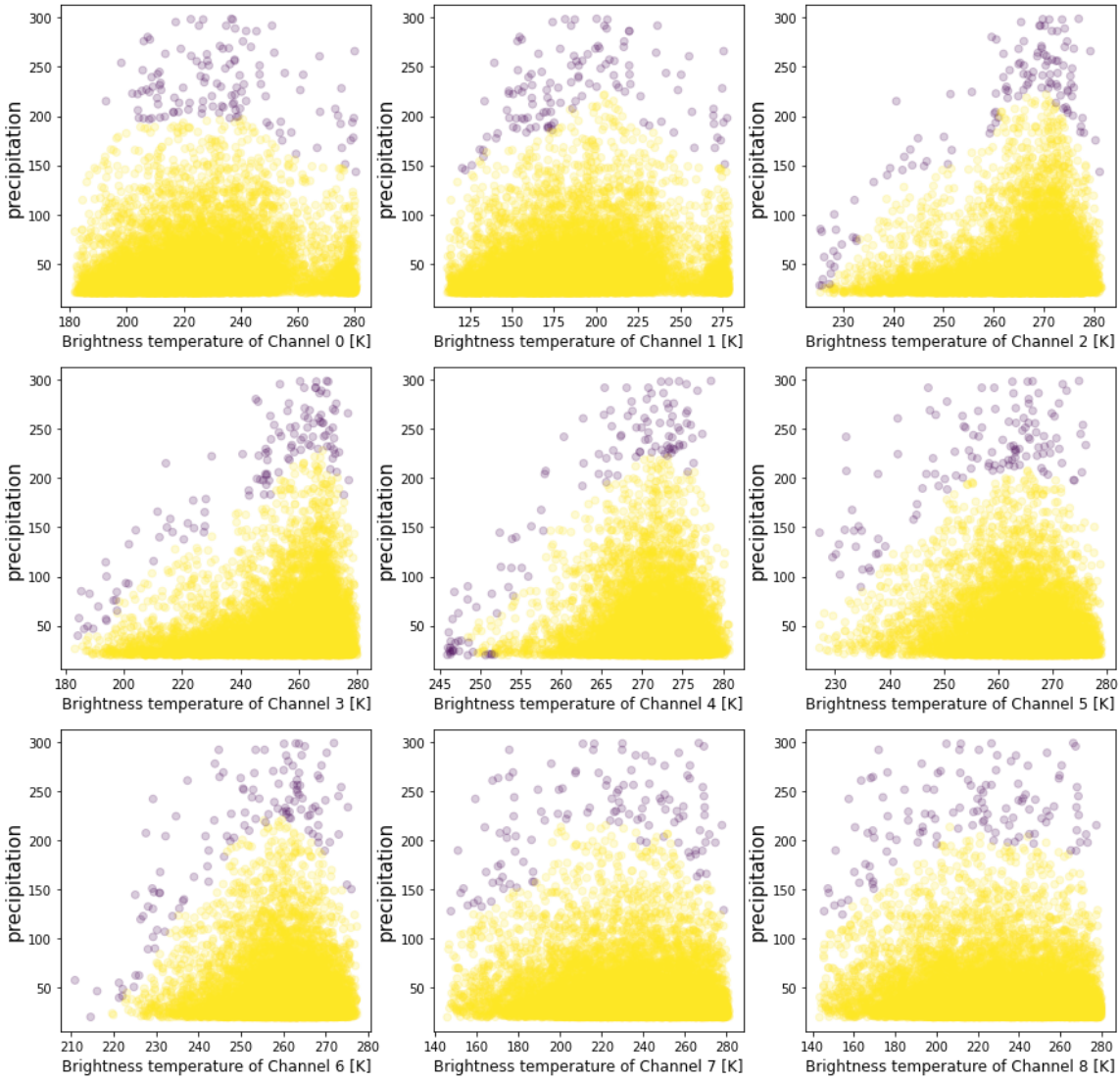


[그림 9.] 설명변수와 반응변수의 산점도(scatter plot)

b) Outlier(이상치) 탐색

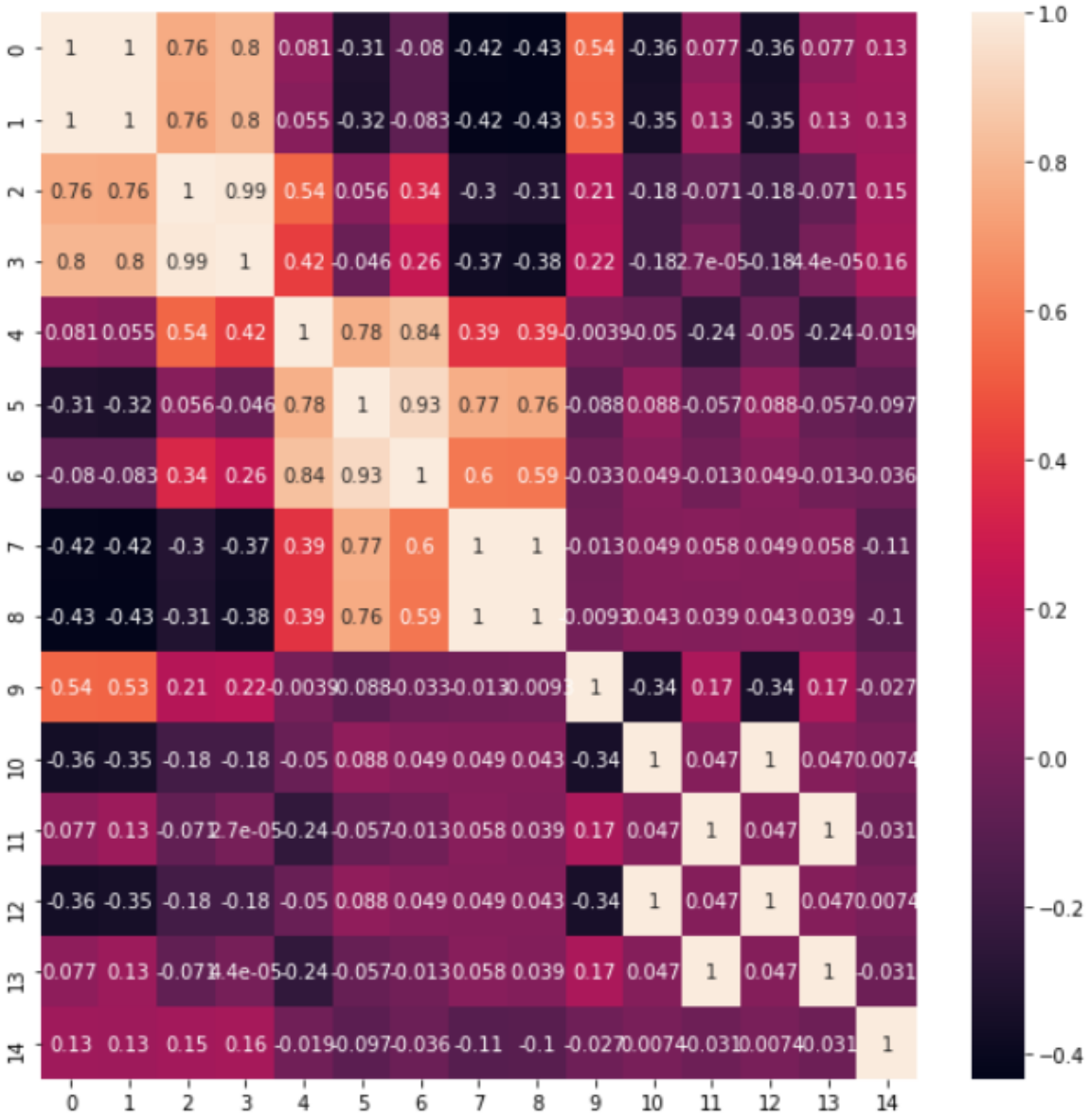
‘9개의 채널(설명 변수)’과 ‘강수량(반응 변수)’의 산점도를 보면 데이터가 주로 분포하는 곳에서 동 떨어진 데이터가 존재한다. 다음과 같은 방법으로 이상치를 탐색하였다.

1. 센서의 특성상 ‘강수량’의 값이 300mm/h을 넘지 못한다. 또한 기존에 알려진 사실과 다르게 강수량이 299~300mm/h의 영역에 몰려있다. 따라서 299mm/h 이상의 데이터는 데이터 가공 과정에서 생긴 오류로 판단하여 제거한다.
2. Regression tree기반의 Isolation Forest 모델(이상치 비율 = 0.1%)을 이용하여 이상치를 탐색하면 다음의 그림과 같다.(보라색 점: 이상치)



[그림 10.] 설명변수와 반응변수의 산점도(scatter plot)를 활용한 이상치 탐색

3) 상관관계 조사



[그림 11.] 모든 변수의 상관계수를 Heatmap으로 표현

0~8 : 0번 ~8번 채널의 밝기값
9 : land type
10 : GMI 센서 위도
11 : GMI 센서 경도
12 : DPR센서 위도
13 : DPR 센서 경도
14 : 강수량 [mm/h]

0~3 번 채널, 4번 ~8번 채널의 변수가 강한 상관 관계를 갖는다. 또한 GMI센서, DPR센서의 위도와 경도도 서로 강한 상관 관계를 갖는다. 이는 다중 공선성의 문제를 발생시킴으로 선형 회귀 학습에서는 적절히 설명변수를 제거하여 학습하는 것이 필요하다.

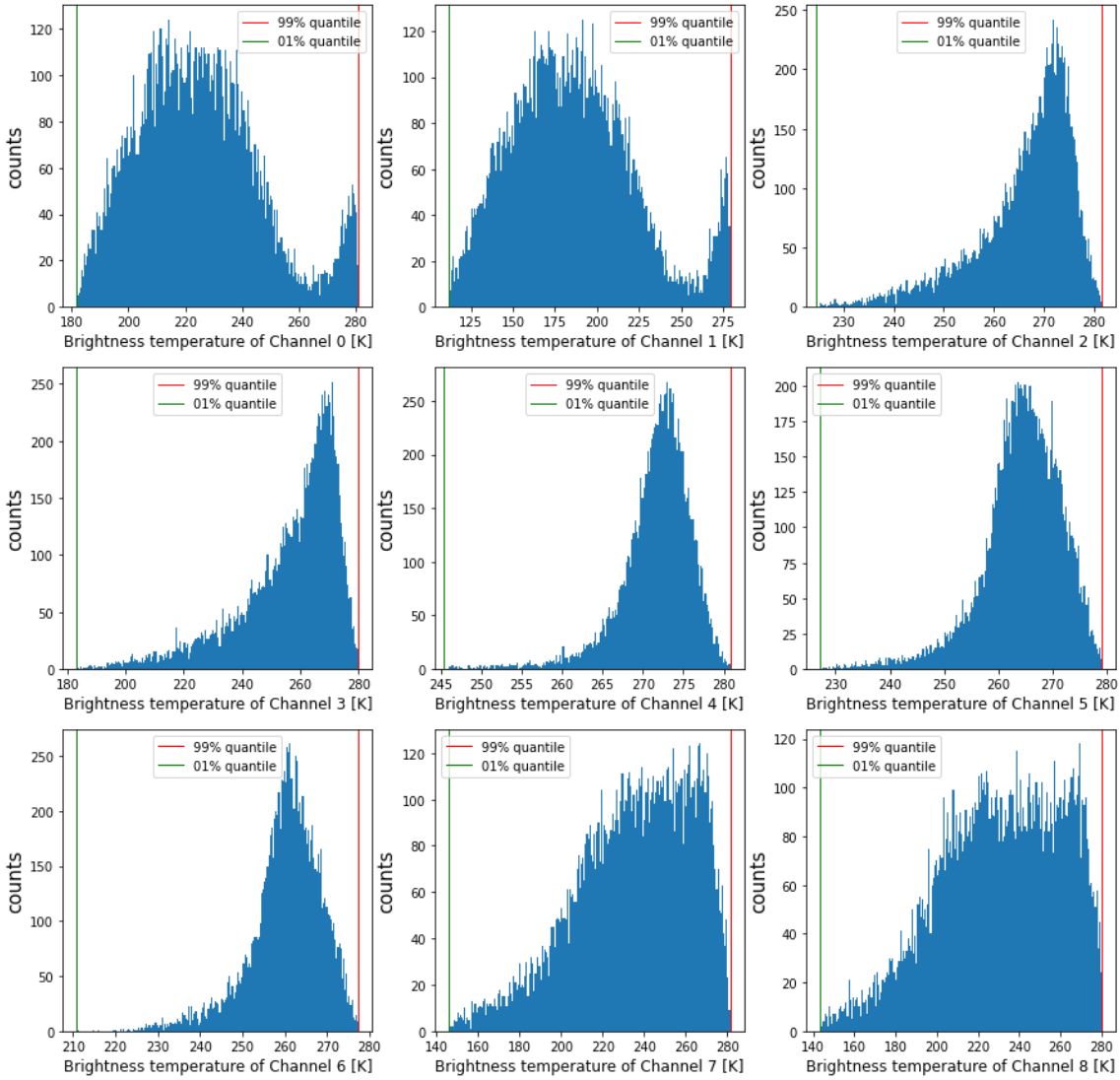
4) 전처리

a) 결측치 처리

전체 데이터에서 결측치는 존재하지 않는다.

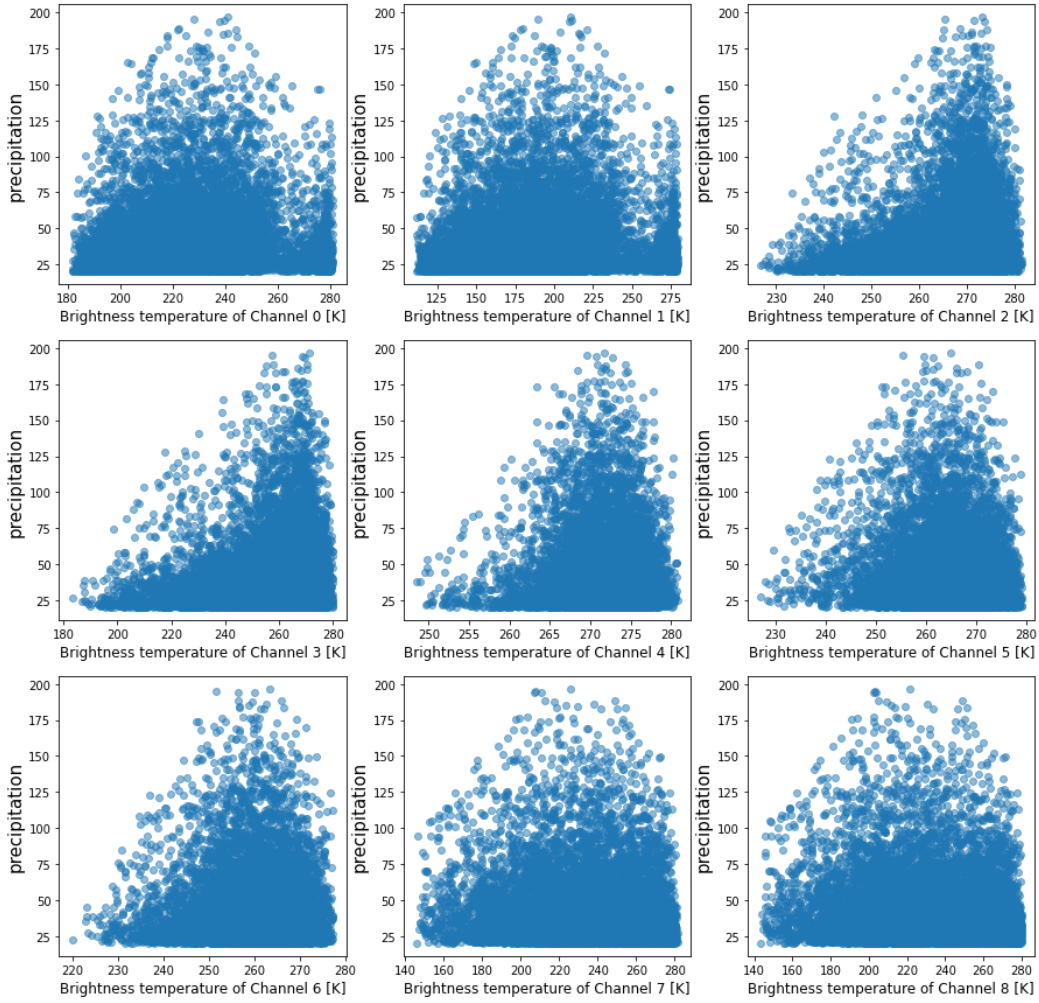
b) 영향관측값(High leverage point) 제거

영향관측값(High leverage point)을 제거한 후 데이터의 분포는 다음과 같다.

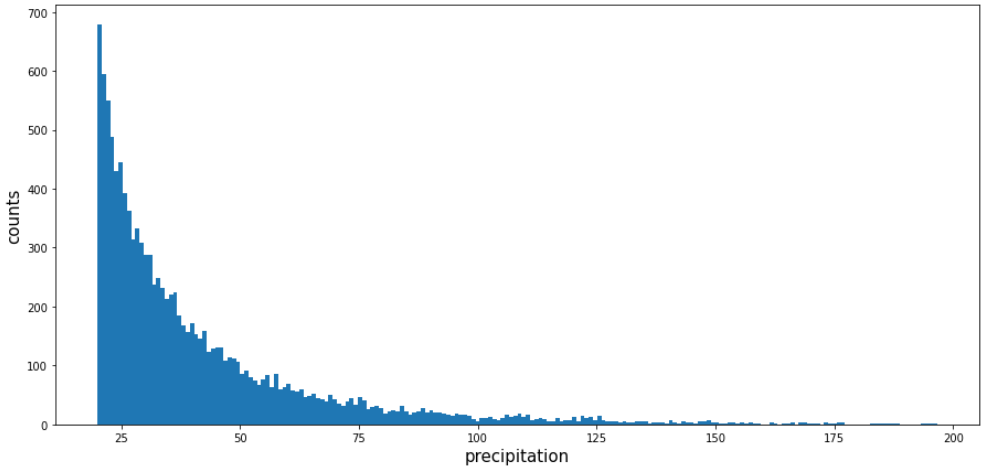


[그림 12.] 영향관측값을 제거한 밝기 온도값의 분포

c) 이상치(Outlier) 제거



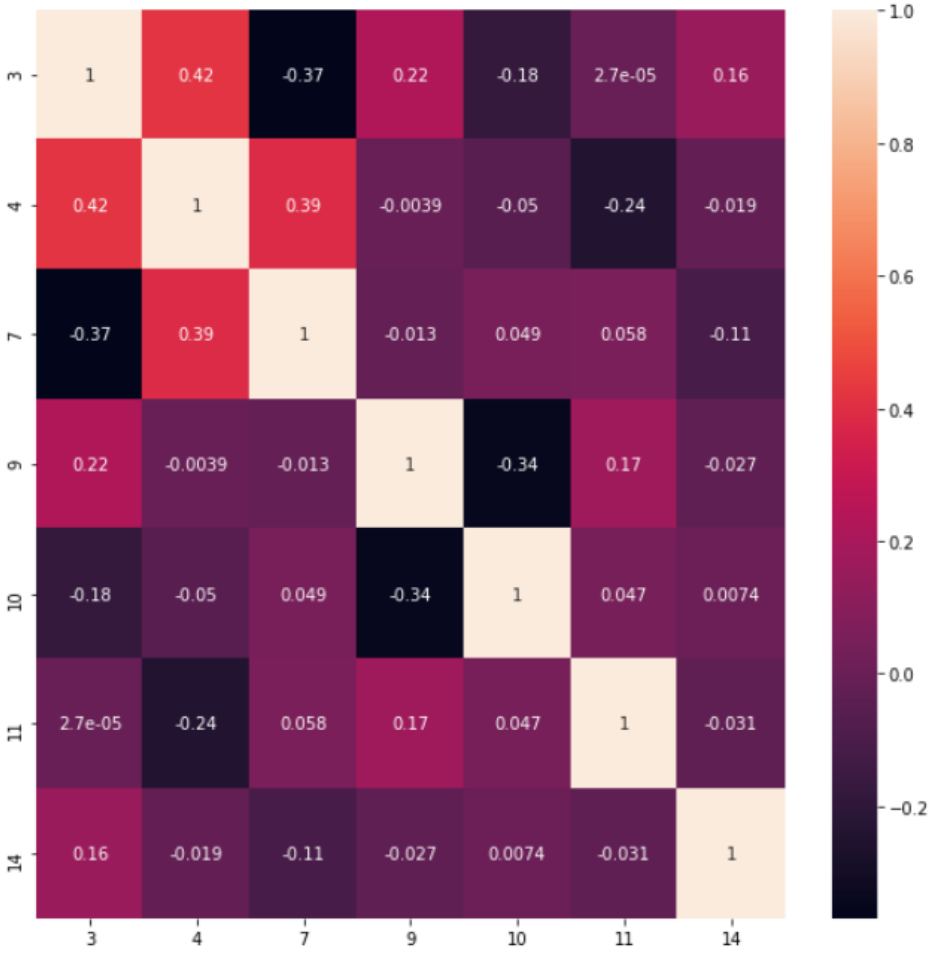
[그림 13.] 이상치를 제거한 설명변수와 반응변수의 산점도(scatter plot)



[그림 14.] 이상치를 제거한 설명변수의 분포

d) 상관관계가 강한 변수 제거

A. 4)를 참고하여 상관 관계가 강한 변수를 제거 하면 다음과 같다.



[그림 15.] 상관관계가 강한 변수 제거 후, 다시 변수별 상관 관계를 Heatmap으로 표현

3,4,7 : 3,4,7번 채널의 밝기값
9 : land type
10 : GMI 센서 위도
11 : GMI 센서 경도
14 : 강수량 [mm/h]

B. 기계학습 모델 1차 성능 비교

1) 선형 회귀 모델

다음과 같은 방법으로 선형 회귀 모델을 훈련하였다.

1. 상관 관계가 강한 변수를 제거한 새로운 데이터(train_select)를 생성한다.
2. train_select를 5개의 fold로 분할한다.
3. scikit-learn의 LinearRegression을 이용하여 각 fold별로 모델을 학습한다
4. Validation fold를 이용하여 CV점수를 구하고 5개의 CV점수를 평균낸다.
5. 데이터를 Normalize한 모델과 그렇지 않은 모델의 평균 CV점수를 비교한다.

계산 결과는 다음과 같다.

```
LinearRegression, normalize = False
단순 선형 회귀 cv_mae_score = 17.420591354370117

LinearRegression, normalize = True
단순 선형 회귀 cv_mae_score = 17.42059326171875
```

	VIF Factor
0	2.544273e+04
1	6.660993e+03
2	1.170341e+05
3	3.581680e+04
4	4.667093e+04
5	7.957601e+04
6	3.999425e+04
7	1.271855e+04
8	1.043093e+04
9	2.149631e+00
10	4.277438e+07
11	2.472574e+06
12	4.238147e+07
13	2.478430e+06

[표 1.] 모든 변수의 VIF Factor

	VIF Factor
0	441.354332
1	929.381566
2	116.195878
3	1.333784
4	68.845156
5	4.506385

[표 2.] 강한 상관관계를 갖는 변수가 제거된 데이터의 VIF Factor

강한 상관관계를 갖는 변수가 제거된 후에도 변수들 사이의 VIF Factor가 매우 큰 값을 갖는다. 이러한 데이터는 다중공선성 문제를 발생시킬 것으로 예상되므로 선형회귀 모델에 적합하지 않다고 판단하였다.

2) 다항 회귀 모델

다음과 같은 방법으로 비선형 다항 회귀 모델을 훈련하였다.

1. train_select를 5개의 fold로 분할한다.
2. degree = 3인 PolynomialFeatures을 이용하여 새로운 feature를 조합한다.
3. scikit-learn의 LinearRegression을 이용하여 각 fold별로 모델을 학습한다
4. Validation fold를 이용하여 CV점수를 구하고 5개의 CV점수를 평균낸다.
5. 데이터를 Normalize한 모델과 그렇지 않은 모델의 평균 CV점수를 비교한다.
6. 3가지 규제 기법(Ridge, Lasso, Elastic)을 사용하여 위의 학습을 다시 반복 비교한다.

학습의 결과는 다음과 같다.

```
Poly_LR_cv_mae, degree  = 3,  normalize = False
Poly_LR_cv_mae = 17.9493408203125

Poly_LR_normal_cv_mae, degree  = 3,  normalize = True
Poly_LR_normal_cv_mae = 17.74848747253418

Poly_Ridge_cv_mae, degree  = 3,  normalize = False
Poly_Ridge_cv_mae = 19.5375919342041

Poly_Ridge_normal_cv_mae, degree  = 3,  normalize = True
Poly_Ridge_normal_cv_mae = 17.378116607666016

Poly_Lasso_cv_mae, degree  = 3,  normalize = False
Poly_Lasso_cv_mae = 17.245738983154297

Poly_Lasso_normal_cv_mae, degree  = 3,  normalize = True
Poly_Lasso_normal_cv_mae = 18.111082077026367

Poly_ElasticNet_cv_mae, degree  = 3,  normalize = False
Poly_ElasticNet_cv_mae = 17.245868682861328

Poly_ElasticNet_normal_cv_mae, degree  = 3,  normalize = True
Poly_ElasticNet_normal_cv_mae = 18.111082077026367
```

[표 3.] 회귀 모델의 훈련 결과

Lasso 규제화를 적용한 degree = 3의 Poly커널 regressor 모델에서 가장 좋은 성능을 보인다.

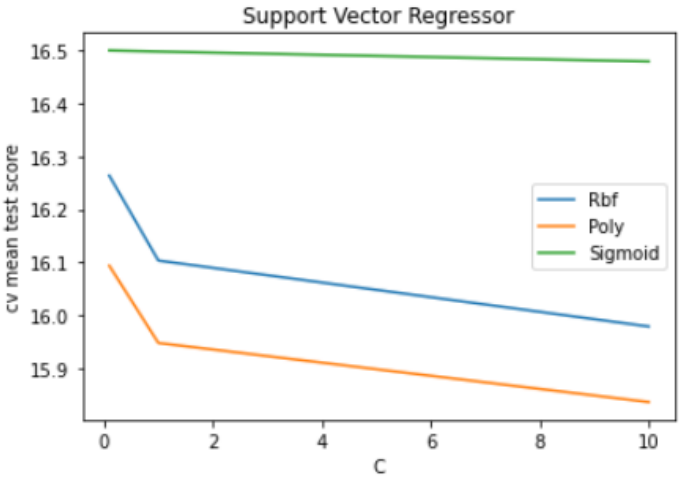
3) Support Vector Machine

다음과 같은 방법으로 서포트벡터 머신 모델을 훈련하였다.

1. 그리드서치를 이용하여 가장 최적화 된 Support Vector Regressor의 parameter(C의 크기, kernel의 종류)를 찾아낸다.
2. train_select를 5개의 fold로 분할한다.
3. scikit-learn의 SVR을 이용하여 1차 그리드서치에서 찾은 C와 kernel을 적용하여 각 fold별로 모델을 학습한다.
4. Validation fold를 이용하여 CV점수를 구하고 5개의 CV점수를 평균낸다.
5. 1차 그리드서치 결과를 바탕으로 더욱 적합한 parameter를 찾기 위한 2차 그리드서치를 진행한다.
6. 2차 그리드서치에서 새롭게 찾은 C를 적용하여 각 fold별로 모델을 학습한다.
7. Validation fold를 이용하여 CV점수를 구하고 5개의 CV점수를 평균낸다.
8. 데이터를 Normalize한 모델과 그렇지 않은 모델의 평균 CV점수를 비교한다.

학습의 결과는 다음과 같다.

```
[{'C': 0.1, 'kernel': 'rbf'},
 {'C': 0.1, 'kernel': 'poly'},
 {'C': 0.1, 'kernel': 'sigmoid'},
 {'C': 1, 'kernel': 'rbf'},
 {'C': 1, 'kernel': 'poly'},
 {'C': 1, 'kernel': 'sigmoid'},
 {'C': 10, 'kernel': 'rbf'},
 {'C': 10, 'kernel': 'poly'},
 {'C': 10, 'kernel': 'sigmoid'}]
```



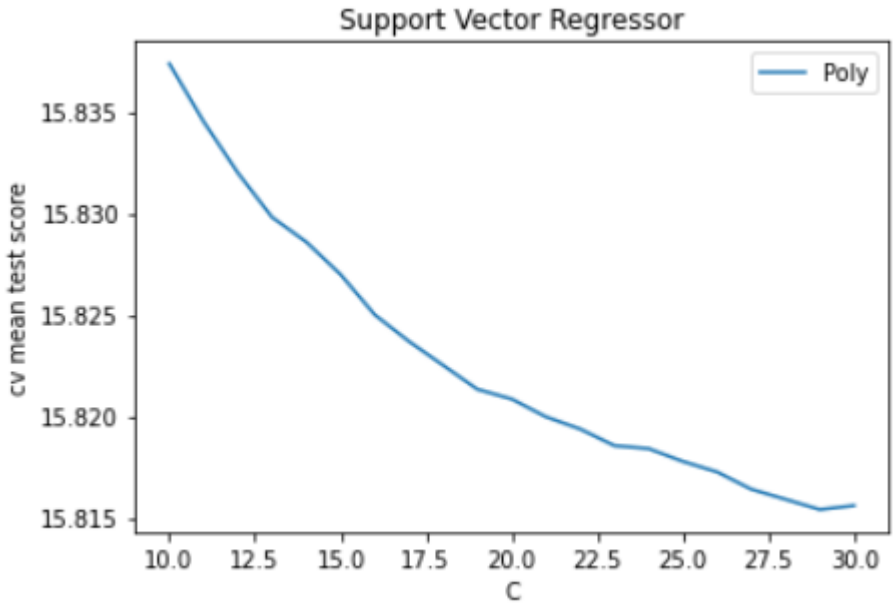
[그림 16.] 1차 그리드서치 결과 (C:0.1, 1, 10)

사용한 kernel의 종류가 Poly일 때 가장 낮은 mae점수를 보인다. Poly를 커널로 사용했을 때, C의 크기가 커질 수록 CV test 평균 mae 점수가 작아진다. 다음의 모수를 가질 때 Support Vector Regressor의 성능이 가장 좋다.

```
SVR, C = 10, kernel = poly
SVR cv_mae_score = 15.837368932183818
```

1차 그리드서치를 바탕으로 다항 함수를 커널로 사용하는 것이 가장 좋은 성능을 내는 것을 알 수 있다. 그리고 마진의 폭과 반비례하는 파라미터인 C의 크기가 커질수록 mae 점수가 낮아짐을 확인할 수 있다. 즉 C가 10이하일 때 C가 증가할 수록 계속 점수가 작아지기 때문에

최적값을 찾은 것이라고 할 수 없다. 따라서, 10이상에서 다시 탐색하기 위해 2차 그리드서치를 진행했다.



[그림 17.] 2차 그리드서치 결과

C가 10이상일 때 29까지 mae 스코어가 작아지는 경향을 보인다. 즉, C가 29일 때 가장 성능이 좋은 것을 확인할 수 있다.

```
SVR, C = 29, kernel = poly
SVR cv_mae_score = 15.815398319162949
```

위에서 찾은 최적의 파라미터를 사용하여 Normalize한 데이터를 훈련시킨 결과는 다음과 같다.

```
SVR, C = 29, kernel = poly
SVR cv_mae_score = 16.051933968245066
```

이론적으로 Support Vector 기반의 모델은 설명변수의 사이즈에 영향을 많이 받는다. 하지만 본 데이터에 대해서는 변수별 사이즈를 조절한 데이터를 학습하였을 때, 성능이 떨어지는 모습을 보인다.

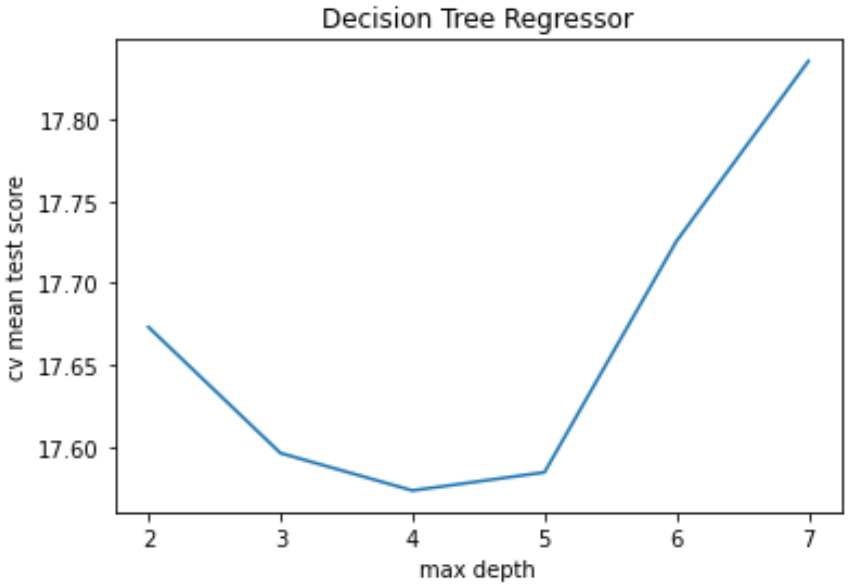
4) 결정 트리

다음과 같은 방법으로 결정 트리 회귀 모델을 훈련했다.

1. GridSearchCV를 이용하여 max_depth 하이퍼 파라미터에 대해서(2,3,4,5,6,7) 가장 좋은 성능의 max_depth를 찾는다.
2. train_select를 5개의 fold로 분할한다.
3. scikit-learn의 DecisionTreeRegressor을 이용하여 그리드서치에서 찾은 max_depth를 적용하여 각 fold별로 모델을 학습한다.
4. Validation fold를 이용하여 CV점수를 구하고 5개의 CV점수를 평균낸다.
5. 데이터를 Normalize한 모델과 그렇지 않은 모델의 평균 CV점수를 비교한다.
6. 모델에 대해 중요도가 높은 feature를 순서대로 선택하여 훈련하고 feature를 추가할 때 마다 CV 점수를 비교한다.
7. 결정 트리 기반 앙상블 모델 3가지(Random Forest, Gradient Boosting, XGBoost)을 사용하여 위의 학습을 다시 반복 비교한다. 이 때 XGBoost에서는 추가적으로 2차 그리드서치를 수행한다.

a) Decision Tree Regressor

```
param_grid = {'max_depth' : range(2,8)}
```



[그림 18.] 결정트리 그리드서치 결과

결정트리의 최대 깊이를 제한하는 파라미터인 max depth가 4일 때 가장 낮은 MAE score를 보였다.

```
Decision Tree Regressor, max_depth = 4
Decision Tree Regressor cv_mae_score = 17.573361979662952
```

위에서 찾은 최적의 파라미터를 사용하여 Normalize한 데이터를 훈련시킨 결과는 다음과 같다.

```
Decision Tree Regressor, max_depth = 4
Decision Tree Regressor cv_mae_score = 17.573361979662952
```

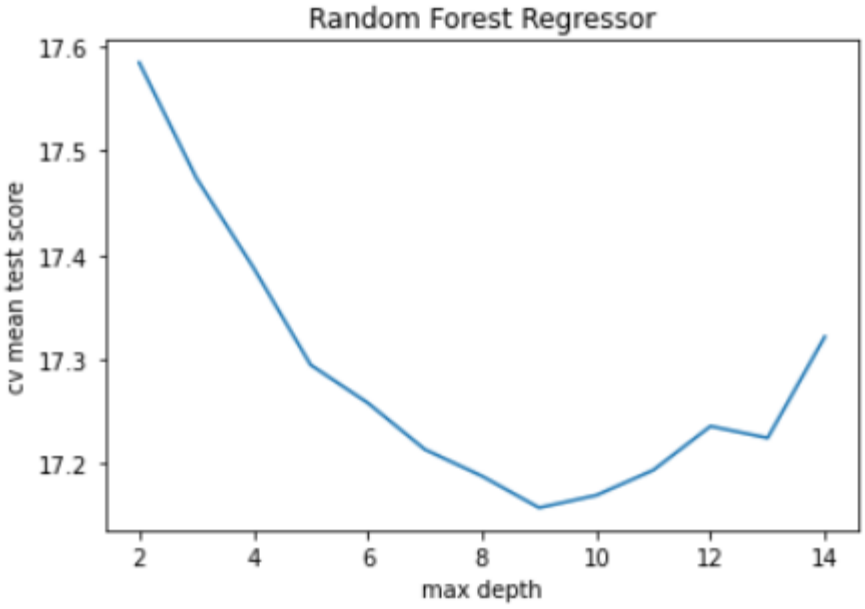
결정트리 모델에서는 Normalize한 데이터와 그렇지 않은 데이터의 스코어가 동일하게 나타났다. 결정 트리 모델의 CART 알고리즘은 변수별 사이즈에 민감하지 않다는 이론적인 내용과 실험 결과가 일치한다.

다음은 결정 트리 기반 앙상블 모델 3가지를 이용하여 데이터를 훈련시켰다.

b) Random Forest

배깅을 적용한 결정 트리 앙상블인 Random Forest 모델을 이용하여 훈련시킨 결과이다. GridSearchCV를 이용하여 Random Forest의 max depth를 [2,3,4,5,6,7,8,9,10,11,12,13,14]의 범위 안에서 탐색하였다.

```
param_grid = {'max_depth' : range(2,15)}
```



[그림 19.] Ranom Forest 그리드서치 결과

결정트리의 최대 깊이를 제한하는 파라미터인 max depth가 9일 때 가장 낮은 mae 스코어가 나타났기 때문에 가장 좋은 성능을 보였다.

```
Random Forest Regressor, max_depth = 9
Random Forest Regressor cv_mae_score = 17.154415814598075
```

위에서 찾은 최적의 파라미터를 사용하여 Normalize한 데이터를 훈련시킨 결과는 다음과 같다.

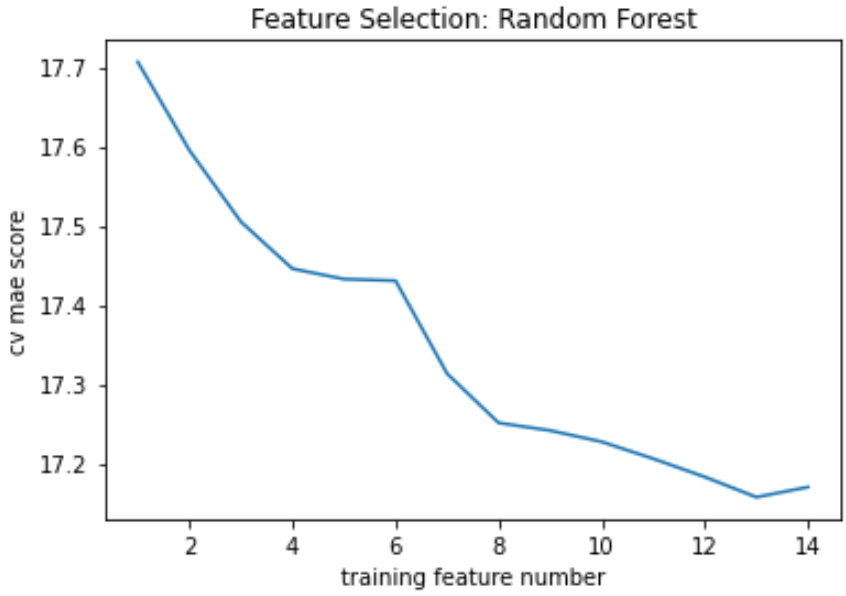
```
Random Forest Regressor, max_depth = 9
Random Forest Regressor Normalization cv_mae_score = 17.171434609067752
```

Random Forest 모델에서는 Normalize한 데이터의 스코어가 더 낮게 나타났다. 하지만 거의 동일한 결과이다.

위에서 찾은 최적의 파라미터를 사용한 모델의 feature importance를 순서대로 정렬한 후, 가장 높은 중요도를 갖는 feature부터 순차적으로 학습시킨 결과는 다음과 같다.

0	4	3	1	2	13	12	8	11	10	5	7	6	9

[표 4.] max depth=9일 때 Random Forest에서의 feature importance (좌: 가장 중요)



[그림 20.] Random Forest에서의 변수 선택 결과

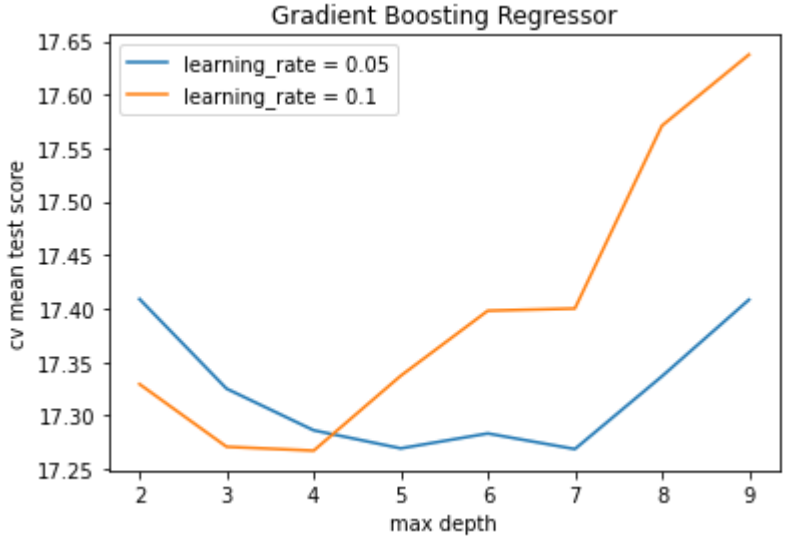
변수 중요도가 가장 높은 channel 0 하나만으로 훈련했을 때 보다 변수를 하나씩 추가할수록 mae 점수가 낮아지는 경향을 보였다. 총 13가지 변수를 선택하여 훈련했을 때 가장 높은 성능을 나타냈다. 즉, 중요도가 가장 낮은 9번째 변수(지표유형)를 제외시키고 훈련하면 성능이 가장 높다.

```
score = 17.14473152898169
```

c) Gradient Boosting Regressor

부스팅을 적용한 결정 트리 앙상블인 Gradient Boosting 회귀 모델을 이용하여 훈련시킨 결과이다.

```
param_grid = {'max_depth': range(2,10),
              'learning_rate': [0.05, .1]}
```



[그림 21.] Gradient Boosting Regressor 그리드서치 결과

learning rate가 0.05일 때보다 0.1일 때 더 최소 mae 점수를 가진다. max depth에 따라서 최적화 된 learning rate의 값이 달라짐을 알 수 있다. 오히려 max depth가 깊어질 수록 learning rate가 0.05일 때 더 좋은 성능을 보이기 때문이다. 그리드서치 결과 데이터에 가장 적합한 파라미터는 max depth=4, learning rate=0.1이다.

```
Gradient Boosting Regressor, max_depth = 4, learning_rate = 0.1
Gradient Boosting Regressor cv_mae_score = 17.271155098545016
```

위에서 찾은 최적의 파라미터를 사용하여 Normalize한 데이터를 훈련시킨 결과는 다음과 같다.

```
Gradient Boosting Regressor, max_depth = 4, learning_rate = 0.1
Gradient Boosting Regressor cv_mae_score = 17.274972810050862
```

Gradient Boosting Regressor 모델에서는 Normalize한 데이터의 스코어가 더 높게 나타났다. 하지만 거의 동일한 결과이다.

위에서 찾은 최적의 파라미터를 사용한 모델의 feature importance를 순서대로 정렬한 후, 가장 높은 중요도를 갖는 feature부터 순차적으로 학습시킨 결과는 다음과 같다.

0	3	4	2	1	8	11	7	13	12	5	10	6	9

[표n.] max depth=4, learning rate=0.1일 때 Gradient Boosting Regressor에서의 feature importance (좌: 가장 중요)

[그림 22.] Gradient Boosting Regressor에서의 변수 선택 결과

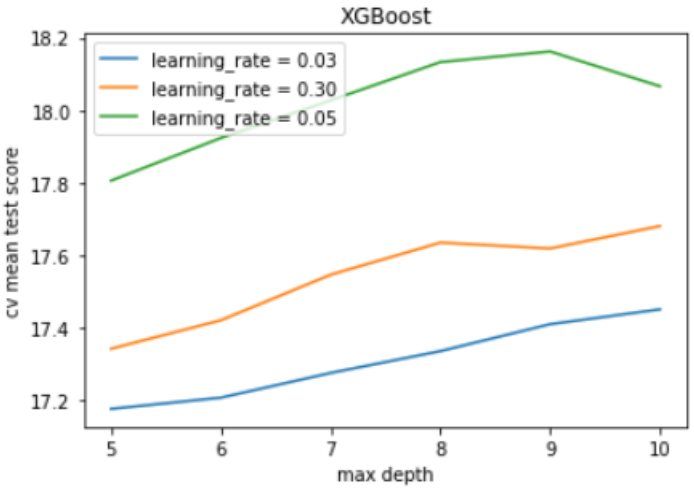
변수 중요도가 가장 높은 channel 0 하나만으로 훈련했을 때 보다 변수를 하나씩 추가할수록 mae 점수가 낮아지는 경향을 보였다. 총 14가지 변수를 모두 선택하여 훈련했을 때 가장 높은 성능을 나타냈다.

```
score = 17.271875702260644
```

d) XGBoost

배깅과 부스팅을 모두 적용한 결정 트리 앙상블인 XGBoost 모델을 이용하여 훈련시킨 결과이다.

```
{'nthread':[4],
 'objective':['reg:linear'],
 'learning_rate': [.03, 0.05, .1],
 'max_depth': [5, 6, 7, 8, 9, 10],
 'min_child_weight': [4],
 'silent': [1],
 'subsample': [0.7],
 'colsample_bytree': [0.7],
 'n_estimators': [500]}
```



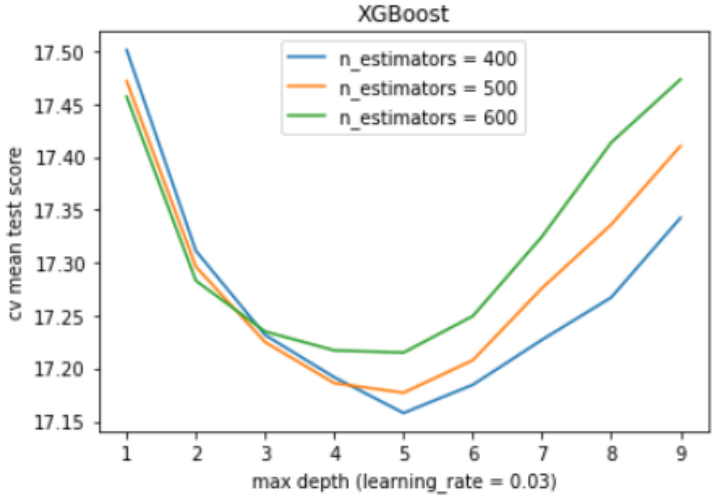
[그림 23.] XGBoost 1차 그리드서치 하이퍼파라미터(좌) 결과(우)

learning rate가 0.03일 때 더 작은 값일 때보다 최소 mae 점수를 가진다. 1차 그리드서치 결과 데이터에 가장 적합한 파라미터와 MAE score는 다음과 같다.

```
XGBoost, learning_rate = 0.03, max_depth = 5
XGBoost cv_mae_score = 17.110212326049805
```

1차 그리드 서치에서 극솟값이 보이지 않아, n_estimator를 조절하여 다시 하이퍼 파라미터의 성능을 탐색하였다.

```
{'nthread':[4],
 'objective':['reg:linear'], #
 'learning_rate': [.03],
 'max_depth': range(1,10),
 'min_child_weight': [4],
 'silent': [1],
 'subsample': [0.7],
 'colsample_bytree': [0.7],
 'n_estimators': [400,500,600]
 }
```



[그림 24.] XGBoost 2차 그리드서치 결과하이퍼파라미터(좌) 결과(우)

파라미터 n_estimators = 400일 때 가장 좋은 성능을 보인다. 2차 그리드서치 결과 데이터에 가장 적합한 하이퍼 파라미터와 MAE 점수는 다음과 같다.

```
XGBoost, learning_rate = 0.03, max_depth = 5, n_estimators = 400
XGBoost cv_mae_score = 17.085966110229492
```

위에서 찾은 최적의 하이퍼 파라미터를 사용하여 Normalize한 데이터를 훈련시킨 결과는 다음과 같다.

```
XGBoost, learning_rate = 0.03, max_depth = 5, n_estimators = 400
XGBoost Normalization cv_mae_score = 17.08957862854004
```

XGBoost 모델에서는 Normalize한 데이터로 학습할 때 성능이 더 낮게 나타났다. 하지만 두 모델에서 큰 차이는 없다.

e) 전진선택법을 활용한 Feature selection (XGBoost)

GridSearchCV를 이용하여 XGBoost에서 최적의 하이퍼 파라미터를 찾았다. 이 모델로 학습한 데이터의 Feature importance를 구할 수 있다. Feature importance 순서대로 정렬한 후, 가장 높은 중요도를 갖는 feature부터 순차적으로 학습시킨 결과는 다음과 같다.

3	9	12	5	4	11	8	1	10	0	6	2	13	7

[표 5.] max depth=6, n_estimator=50일 때 XGBoost에서의 feature importance (좌: 가장 중요)

0~8 : 0번 ~8번 채널의 밝기값
9 : land type
10 : GMI 센서 위도
11 : GMI 센서 경도
12 : DPR센서 위도
13 : DPR 센서 경도
14 : 강수량 [mm/h]

[그림 25.] XGBoost에서의 변수 선택 결과

총 12가지 변수를 선택하여 훈련했을 때 가장 높은 성능을 나타냈다. 즉, 상대적으로 중요도가 낮은 channel 7 센서값 DPR 센서 경도 변수를 제거하고 훈련했을 때 더 좋은 점수를 얻는다.

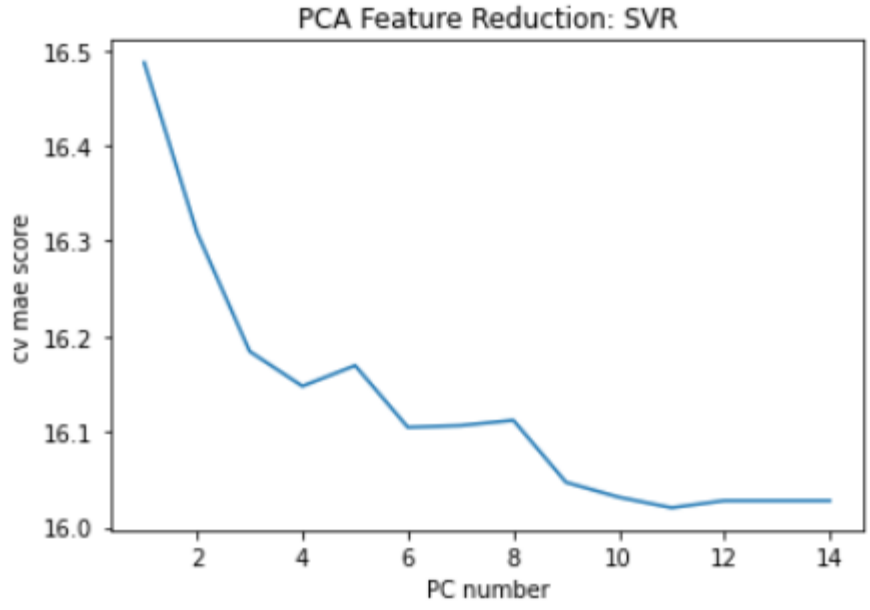
```
Feature Selection of XGBoost
cv mae score: 17.08363
```

5) PCA를 통한 변수 선택

지금까지 선형회귀, SVR, 결정나무 및 다양한 커널 규제 기법을 적용하여 모델을 훈련하였다. 그 중에서도 가장 성능이 좋았던 SVR 모형에서 PCA를 통한 차원축소 기법을 적용해 보고자 한다.

1. sklearn의 PCA를 활용하여 데이터의 14개 PC를 모두 계산한다.
2. 전진선택법의 방식으로 제1 주성분 부터 차례로 주성분을 추가한다.
3. 주성분을 추가할 때마다 SVR모델의 cross validation MAE score를 계산한다.

위와 같은 방법으로 SVR 모형에서 PCA를 통한 차원축소 기법을 적용하면 다음과 같다.



[그림 26.] 주성분 개수에 따른 스코어

제11 주성분까지 추가 했을 때 가장 성능이 좋았다.

```
Feature Selection of SVR
cv mae score: 16.020509257989527
```

원래 데이터로 학습시 cv mae score가 15.815임을 고려하면, PCA를 통해서 차원을 축소한 후 SVR을 훈련하는 것 보다 원래의 데이터를 학습할 때 성능이 좋다.

6) 성능 비교

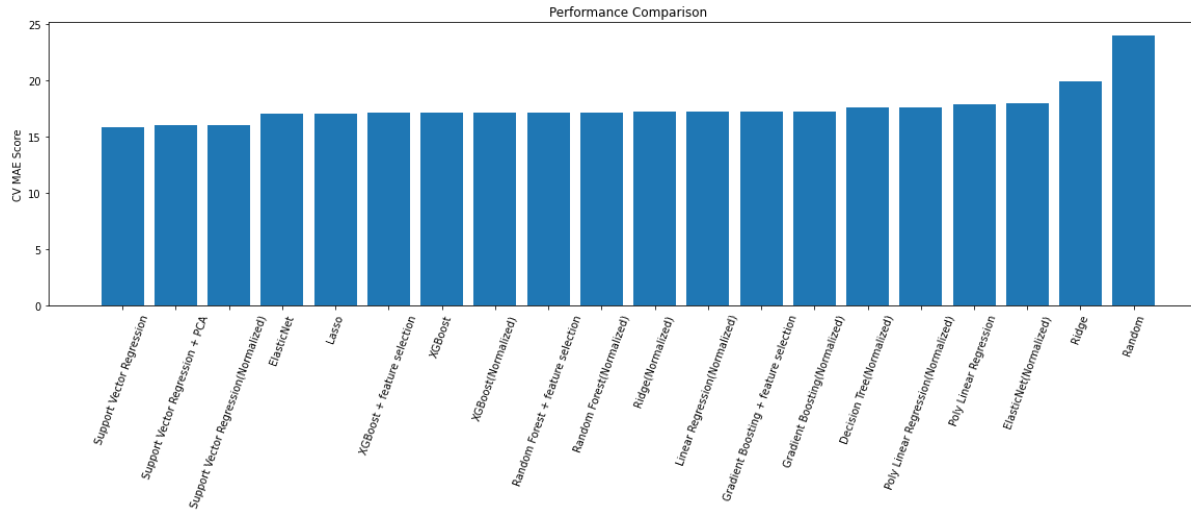
모델	Features	규제	Normalization	cv MAE
무작위 예측	normal	No	No	23.9522
선형회귀	normal	No	No	17.226
	normal	No	Yes	17.226
비선형 회귀	PolynomialFeatures(degree=3)	No	No	17.824
	PolynomialFeatures(degree=3)	No	Yes	17.623
	PolynomialFeatures(degree=3)	Ridge	No	19.872
	PolynomialFeatures(degree=3)	Ridge	Yes	17.173
	PolynomialFeatures(degree=3)	Lasso	No	17.052
	PolynomialFeatures(degree=3)	Lasso	Yes	17.937
	PolynomialFeatures(degree=3)	ElasticNet	No	17.051
	PolynomialFeatures(degree=3)	ElasticNet	Yes	17.937
SVR	PolynomialFeatures(degree=3)	No	No	15.815
	PolynomialFeatures(degree=3)	No	Yes	16.052
	PolynomialFeatures(degree=3), PCA	No	No	16.020
Decision Tree	normal	No	No	17.573
	normal	No	Yes	17.573
Random Forest	normal	No	No	17.154
	normal	No	Yes	17.171
	feature selection	No	No	17.154
Gradient Boosting	normal	No	No	17.259
	normal	No	Yes	17.26
	feature selection	No	No	17.259
XGBoost	normal	No	No	17.086
	normal	No	Yes	17.089
	feature selection	No	No	17.084

cv MAE 점수가 15.815로 가장 성능이 높은 Support Vector Regression의 최종 test score는 다음과 같다.

```
SVR, C = 29, kernel = poly
SVR final test mae score = 14.898038797651967
```

이는 CV score보다 더 낮은 값으로 오히려 기존 train data보다 test data에 더 적합한 모델로 학습한 것을 알 수 있다.

III. 결론



[그림27.] CV 점수가 가장 좋은 모델부터 순차적으로 표시한 막대 그래프

CV score를 바탕으로 강수량 예측을 위해 사용한 데이터에 가장 적합한 기계학습 모델은 Support Vector Regressor이다. 그리드서치를 통해 찾아낸 최적의 파라미터는 C=29, kernel=poly이다. 처음에 분리시켜 놓은 Test data에 대해 MAE score를 계산한 결과는 CV score보다 더 좋은 값인 14.8980이다. 또한 Ridge 회귀 모델을 제외한 나머지 모델에서는 Normalize 전후 훈련 결과 차이가 거의 없음을 알 수 있다.

[고찰]

1) 지형변수의 feature importance가 굉장히 낮게 나타났다. 이를 Ocean, Land, Coastal, Inland Water 이렇게 네 가지의 dummy 변수로 바꿔준다면 성능의 향상을 기대할 수 있을 것으로 보인다.
2) 비교적 분포가 일정하고 강수량 예측에 중요도가 떨어지는 위도, 경도 변수들을 제외하고 훈련한 후 이를 비교했으면 차원이 줄어들어 더 좋은 결과를 얻었을 것으로 추정된다.
3) 이미지 데이터를 픽셀 단위로 flatten하여 분석했기 때문에 픽셀 주변 이미지에서 얻을 수 있는 정보를 손실했다. 특히 온도는 주변 기후와 밀접한 연관성을 보이기 때문에 이렇게 이미지에서만 확인할 수 있는 중요한 정보를 손실한 것이다. 본 레포트에서는 이미지의 온전한 정보를 활용할 수 없었기 때문에 성능이 제한적으로 나타난 것으로 보인다.

<참고 자료>

[과소표집]
https://en.wikipedia.org/wiki/Undersampling
[데이터 출처]
https://dacon.io/competitions/official/235591/data/
[데이콘 대회 데이터 설명 ppt 링크]
https://www.slideshare.net/daconist/ai-2-230847126?ref=https://dacon.io/competitions/official/235591/talkboard/400589
[데이콘 대회 데이터 설명 동영상]
https://www.youtube.com/watch?v=sZqQlWIIG_s&feature=youtu.be