선형 회귀를 이용한 데이터 분석

날씨 정보로 모기 개체 수 예측하기

학번	학과	이름
2017010698	수학과	오서영

목차

Ⅰ. 문제 정의

1. 목적 및 필요성

모기는 일본뇌염, 말라리아와 같은 다양한 질병의 매개체 역할을 한다. 이러한 질병의 감염자 수는 꾸준히 증가하고 있는 추세이고, 이는 모기 개체 수의 증가와 무관하지 않다. 그렇기에 사람들의 안전한 생활을 위해서, 모기 개체수를 예측하는 것은 필요한 일이라고 생각한다.

2. 문제 정의

모기의 개체 수를 모기와 관련된 데이터가 아닌 다른 얻기 쉬운 외부 데이터를 통해 예측하고자 한다. 기상청을 통해 누구나 쉽게 얻을 수 있는 날씨 데이터를 선형회귀 모델의 독립변수로 사용할 것이다.

모기의 서식환경에 가장 크게 영향을 미치는 날씨요소는 기온과 습도이다. 실제로 부산지방기상청에서 진행한 연구에 따르면, 기온과 강수량 그리고 습도는 모기의 활동과 어느 정도 상관관계를 가진다고 한다. 이를 토대로 기온과 습도를 독립변수로, 모기 개체 수를 종속변수로 가지는 여러 선형 회귀 모델을 구현하고자 한다.

Ⅱ. 데이터 준비

1. 독립변수(X)

'기상자료개방포털'에 공개된 서울시의 2019년 4월 16일부터 10월 31일 사이의 평균 기온, 평균 상대 습도 데이터를 사용했다.

날짜	지점	평균기온(℃	최저기온(℃	최고기온(℃)	지점	지점명	일시	평균 상대습도(%)
2019-04-16	108	14.9	6.8	22.6	108	서울	2019-04-16	26.3
2019-04-17	108	15.3	9.1	21.8	108	서울	2019-04-17	37.5
2019-04-18	108	12.6	10.4	16.2	108	서울	2019-04-18	74.4
2019-04-19	108	13.8	9.6	19.7	108	서울	2019-04-19	57.5
2019-04-20	108	14.3	9.7	19.1	108	서울	2019-04-20	66.9
2019-04-21	108	14.8	12.2	20.2	108	서울	2019-04-21	77.8
2019-04-22	108	19.3	10.7	28.2	108	서울	2019-04-22	50.6
2019-04-23	108	20.4	15.3	25.5	108	서울	2019-04-23	52
2019-04-24	108	19.5	15.8	25.9	108	서울	2019-04-24	70.4

< 그림 2-1. 서울시 평균 기온, 평균 상대 습도 데이터 일부 >

2. 종속변수(y)

서울시에서는 디지털 모기 측정기(DMS)로 얻은 데이터를 이용하여 모기 예보제를 시행하고 있다. 이를 통해 공개된 '2019년 서울시 디지털 모기측정기(DMS) 채집모기 현황(2019.4.15 ~ 2019.10.31.)' 데이터를 종속변수로 사용했다.

DMS 포집내역 (2019.4.15 ~ 2019.10.31)

	포집량	모기	기타
계	490,101	484,586	5,517
2019-04-15	-	-	-
2019-04-16	48	48	-
2019-04-17	219	218	1
2019-04-18	321	314	7
2019-04-19	242	242	-
2019-04-20	471	468	3
2019-10-28	268	266	2
2019-10-29	440	439	1
2019-10-30	210	206	4
2019-10-31	185	185	-

< 그림 2-2. 서울시 DMS 채집 모기 현황 데이터 일부 >

Ⅲ. 구현 및 실험

1. 다중선형회귀

```
import numpy as np
import matplotlib.pylab as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

- 필요한 패키지를 불러온다

```
# independent variable : shape of [Tempurature, Humidity]
X = np.array([[14.9, 26.3],[15.3, 37.5],[12.6, 74.4],[13.8, 57.5],[14.3, 66.9],[14.8, 77.8],[19.3, 50.6],[20.4, 52.0],[19.5, 70.4],
 [13.2, 75.9],[8.4, 82.1],[11.6, 52.6],[12.9, 48.8],[13.0, 52.9],[15.5, 52.8],[16.4, 48.9],[17.1, 28.5],[17.9, 37.5],[19.5, 40.4],
 [19.2, 35.8],[14.5, 27.1],[15.0, 33.4],[15.6, 48.3],[16.6, 43.3],[17.8, 48.0],[20.3, 39.0],[21.6, 36.0],[18.7, 43.4],[19.6, 44.5],

# dependent variable
y = np.array([48,218,314,242,468,506,603,814,771,1107,104,101,894,782,705,1345,1507,1258,1137,
              1315,1108,1172,1031,1169,1300,1636,1633,1543,1364,1395,1421,1629,1531,1697,1698,
```

- 데이터 셋을 생성한다. X 데이터는 [평균 온도, 평균 상대 습도] 형태로 이루어진다.

```
# Explore dataset
print("shape of X :", X.shape)
print("shape of y :", y.shape)
```

```
shape of X : (199, 2)
shape of y : (199,)
```

- 약 200개의 데이터로 이루어져 있다.

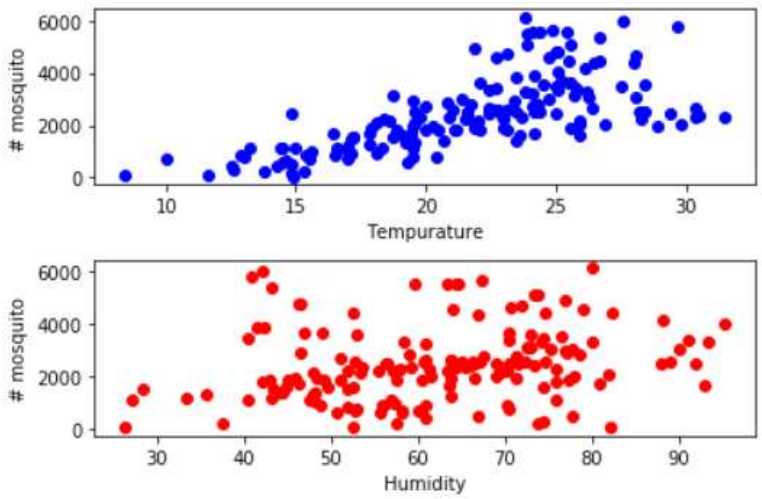
```
# split train and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1004)
```

- 만든 데이터 셋을 훈련데이터와 테스트데이터로 나눈다.

```
# Visualization
fig = plt.figure()
p1 = fig.add_subplot(2, 1, 1)
p2 = fig.add_subplot(2, 1, 2)

p1.plot(X_train[:,0], y_train, 'bo')  # Tempurature
p1.set_xlabel("Tempurature")
p1.set_ylabel("# mosquito")
p2.plot(X_train[:,1], y_train, 'ro')  # Humidity
p2.set_xlabel("Humidity")
p2.set_ylabel("# mosquito")
plt.tight_layout()

plt.show()
```



- 모델을 구현하기 전에, 독립변수 각각을 종속변수와 시각화 해봤다.
- '모기개체수~평균온도'를 시각화 한 결과가 더 선형적으로 보이기에 둘의 상관관계가 더 높을 것이라 가늠할 수 있다.

```
model = LinearRegression()
model.fit(X_train, y_train)
```

- 다중 선형 회귀 모델을 구현한다.

2. 다항회귀

```
# independent variable : shape of [Tempurature]
X = np.array([[14.9],[15.3],[12.6],[13.8],[14.3],[14.8],[19.3],[20.4],[19.5],[13.2],[ 8.4],[11.6],[12.9],[13. ],[15.5],[16.4],[17.1],
       [19.5],[19.2],[14.5],[15. ],[15.6],[16.6],[17.8],[20.3],[21.6],[18.7],[19.6],[20.7],[23.5],[23.6],[22.8],[19.4],[16.4],[17.2],
```

- 두번째로는 상대적으로 상관관계가 더 높아 보였던
평균온도 데이터만 사용하여 다항회귀 모델을 구현했다.

```
poly = PolynomialFeatures(degree = 3, include_bias = False)
X_poly = poly.fit_transform(X_train)
model = LinearRegression()
model.fit(X_poly, y_train)
```

- 3차 다항회귀 모델을 구현한다.

IV. 결과 분석

1. 다중선형회귀

```
print("W =", model.coef_)
print("b =", model.intercept_)
```

```
W = [[198.05976766   8.13000628]]
b = [-2330.12148808]
```

- 모델을 통해 학습한 가중치와 편향은 다음과 같다
- 예측값은
"(198.05976766)*(평균온도) + (8.13000628)*(평균 상대 습도) + (-2330.12148808)"
식을 통해 계산된다.

```
# prediction of one data
W = model.coef_
b = model.intercept_

a_data = np.array([30, 60])
a_pred = np.dot(W,a_data.T) + b
print("a prediction of # mosquito is", a_pred)
```

```
a prediction of # mosquito is 4099.47191818556
```

- 평균온도가 30도이고 평균상대습도가 60%일 때,
모기 개체수를 예측한 결과이다.

```
train_pred = model.predict(X_train)
test_pred = model.predict(X_test)
```

```
print('Train MAE :', metrics.mean_absolute_error(y_train, train_pred))
print('Test MAE :', metrics.mean_absolute_error(y_test, test_pred))
```

```
Train MAE : 769.159651291196
Test MAE : 804.6730773077359
```

```
print("Train R2_Score:", metrics.r2_score(y_train, train_pred))
print("Test R2_Score:", metrics.r2_score(y_test, test_pred))
```

```
Train R2_Score: 0.4646285517625959
Test R2_Score: 0.4437745857207037
```

- MAE 값과 R2 값을 통해 모델을 평가했다.
- 오버피팅이 심하진 않지만 MAE 값이 크고, R2값이 작기 때문에 성능이 좋은 모델은 아니라고 생각한다.

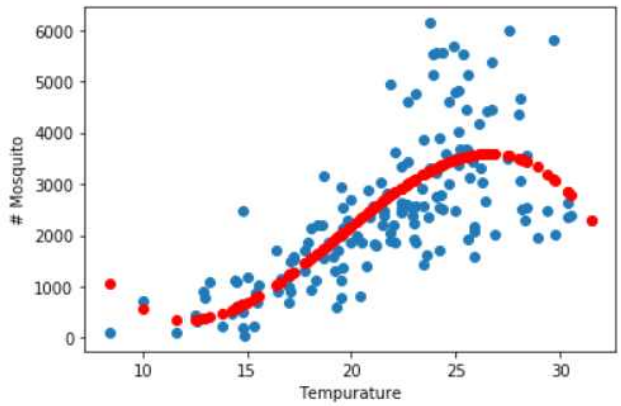
2. 다항회귀

```
print("W =", model.coef_)
print("b =", model.intercept_)
```

```
W = [[-2016.95290956   121.32301383    -2.08791732]]
b = [10691.7202389]
```

- 예측값은
"(-2016.95290956)*(평균온도) + (121.32301383)*(평균온도)^2 +
(-2.08791732)*(평균온도)^3 + (10691.7202389)"
식을 통해 계산된다.

```
# Visualization
plt.scatter(X_train, y_train)
plt.plot(X_train, train_pred,'ro')
plt.xlabel("Tempurature")
plt.ylabel("# Mosquito")
plt.show()
```



- 실제 값과 예측 값을 시각화 했다

```
print('Train MAE :', metrics.mean_absolute_error(y_train, train_pred))
print('Test MAE :', metrics.mean_absolute_error(y_test, test_pred))
```

```
Train MAE : 715.471579353026
Test MAE : 827.6449448401585
```

```
print("Train R2_Score:", metrics.r2_score(y_train, train_pred))
print("Test R2_Score:", metrics.r2_score(y_test, test_pred))
```

```
Train R2_Score: 0.5478069652841393
Test R2_Score: 0.45754142253537766
```

- 이전의 다중선형회귀 모델보다 성능은 약간 개선됐지만,
오버피팅이 심해졌기 때문에 이 또한 좋은 모델은 아니라고 생각한다.

V. 결론 도출

내가 사용한 데이터는 일 별 평균온도, 평균상대습도, 모기 개체 수였다. 이들은 모두 시간 순서가 있는 데이터였지만, 나는 순서정보를 무시하고 각각의 숫자 값만을 사용하여 모델을 구현했다. 이것이 내가 생각하는 모델 성능이 낮은 가장 큰 이유이다. 만약 시계열 예측이나 RNN같은 모델을 구현했다면 더 좋은 결과가 나올지도 모르겠다. 또한 모기 성충의 수명이 10일 전후라고 하는데 이러한 배경정보를 모델에 반영하지 못한 것도 또 다른 이유인 듯 싶다.

VI. 참고자료

[1] [여름철 모기 예측] 온도, 습도로 여름철 불청객 모기를 예측하다?, "온도,습도,모기", http://blog.naver.com/PostView.nhn?blogId=kma_131&logNo=221066015623, (2020.06.11.)

[2] 기온 및 강수량 변화와 모기개체수 http://web.kma.go.kr/notify/press/reqional_list.jsp?bid=press2&mode=view&num=490, (2020.06.11.)

[3] [기상자료개방포털] 기후통계분석 > 통계분석 > 기온분석, https://data.kma.go.kr/stcs/grnd/grndTaList.do?pgmNo=70, (2020.06.11.)

[4] [기상자료개방포털] 데이터 > 기상관측 > 지상 > 종관기상관측(ASOS), https://data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36, (2020.06.11.)

[5] 2019년 서울시 디지털 모기측정기(DMS) 채집모기 현황 (2019.4.15 ~ 2019.10.31.), "모기 개체수", http://news.seoul.go.kr/welfare/mos_dmsnblt2, (2020.06.11.)