k-NN 분류를 이용한 데이터 분석

에너지 소비 효율 등급 분류하기

학번	학과	이름
2017010698	수학과	오서영

목차

Ⅰ. 문제 정의

1. 목적 및 필요성

'에너지 소비 효율 등급' 표시제도는 소비자들이 효율이 높은 에너지 절약형 제품을 쉽게 구입할 수 있도록 만들어진 제도이다. 등급은 1~5등급으로 나누어져 표시 되어있고, 등급이 낮을수록 에너지 효율이 높아서, 에너지를 더 절약할 수 있다는 의미를 가진다. 이 제도는 오래전부터 시행됐지만, 어떤 기준으로 등급이 나뉘는지 잘 아는 사람은 드물다. 그렇기에 누구나 등급을 예측해볼 수 있는 분류기를 만들어 보려 한다.

2. 문제 정의

최대 소비 전력량, 시간당 이산화탄소 배출량, 연간 에너지 비용을 독립변수로, 에너지 소비 효율 등급을 종속 변수로 하는 k-NN 분류 모델을 구현하고자 한다.

Ⅱ. 데이터 준비

1. 독립변수(X)

'공공데이터포털'에 공개된 '에너지소비효율등급제도 기기부문 제품정보' 데이터를 사용했다. 주어진 데이터 중 '최대 소비 전력량', '시간당 이산화탄소 배출량', '연간 에너지 비용'을 독립변수로 사용했다. 전체 156377개의 데이터 중, 세 요소가 모두 적절히 공개되어 있는 136195~136394 번째 데이터, 즉 200개의 데이터를 추출했다.

2. 종속변수(y)

종속 변수 또한 '공공데이터포털'에 공개된 '에너지소비효율등급제도 기기부문 제품정보' 데이터를 사용했다. 주어진 데이터 중 '등급'이 1~2일 때 '0', 3일 때 '1', 4~5일 때 '2'로 종속변수를 라벨링했다.

즉 입력 특징은 3개, 출력은 3클래스인 k-NN 분류기를 구현할 것이다.

1	신청번호	품목명	모델명	등급	최대소비전력량	용량	시간당 이산화탄소 배출량(g)	연간 에너지 비용
136195	2.52E+08	전기냉장고	CRFT-D046V	1	18.44	45.8	3	12,000
136196	2.52E+08	전기냉장고	B607W	2	56.61	591.2	15	50,000
136197	2.52E+08	전기냉장고	CRFT-D207V	1	35.69	202.6	9	27,000
136198	2.52E+08	전기냉장고	SR-S10AU	1	20.32	96.4	5	17,000
136199	2.52E+08	전기냉장고	B507W	2	50.16	509.3	14	46,000
136200	2.52E+08	전기냉장고	RF85M9002V	3	64.65	862.9	17	57,000
136201	2.52E+08	전기냉장고	EBB3500MG	3	41.47	343.4	13	44,000
136202	2.52E+08	전기냉장고	F90M1-G	2	32.43	89.1	9	31,000
136203	2.52E+08	전기냉장고	F90M1-G	2	32.43	89.1	9	31,000

< 그림 2-1. 에너지소비효율등급제도 기기부문 제품정보 데이터 일부 >

Ⅲ. 구현 및 실험

1. 데이터 준비하기

```
import numpy as np
import matplotlib.pylab as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

필요한 패키지를 불러온다

```
# independent variable : shape of [최대소비전력량, 시간당 이산화탄소 배출량(g), 연간 에너지 비용]
X = np.array([[18.44, 3, 12000], [56.61, 15, 50000], [35.69, 9, 27000], [20.32, 5, 17000], [50.16, 14, 46000], [64.65, 17, 57000],
              [41.47, 13, 44000], [32.43, 9, 31000], [32.43, 9, 31000], [39.82, 17, 56000], [23.85, 5, 17000], [67.98, 18, 60000],
              [67.98, 18, 60000], [67.76, 18, 60000], [67.98, 18, 60000], [64.55, 17, 56000], [74.73, 21, 69000], [66.4, 19, 62000],
              [20.19, 9, 28000], [50.23, 17, 57000], [19.67, 6, 21000], [19.67, 6, 21000], [19.67, 6, 21000], [19.67, 6, 21000],
              [19.67, 6, 21000], [19.67, 6, 21000], [19.67, 6, 21000], [19.67, 6, 21000], [65.72, 20, 68000], [33.87, 14, 47000],
```

```
# dependent variable : 에너지 소비 효율 등급 (energy consumption efficiency rating)
y = np.array([0, 0, 0, 0, 0, 1, 1, 0, 0, 2, 0, 1, 1, 1, 1, 0, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 2, 0, 1,
              1, 2, 2, 2, 1, 2, 2, 2, 1, 1, 0, 2, 2, 1, 0, 0, 1, 2, 2, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,
              0, 0, 0, 1, 1, 1, 1, 1, 2, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0, 0,
              1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 0, 1, 1, 2, 2, 2, 2, 2, 2, 0, 1, 2, 0, 2, 0, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0,
```

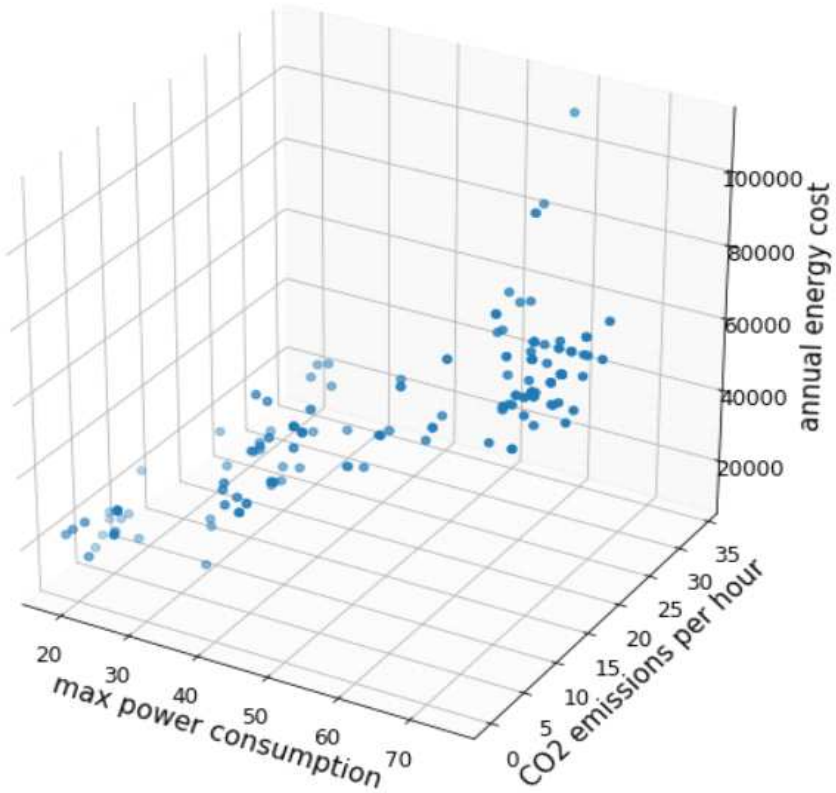
데이터 셋을 생성한다. X 데이터는 [최대소비전력량, 시간당 이산화탄소 배출량, 연간 에너지 비용] 형태로 만들어지고, y 데이터는 0,1,2 세가지로 라벨링된다.

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

```
shape of X : (200, 3)
shape of y : (200,)
```

200개의 데이터로 이루어져 있다.

```
# visualization
plt.rcParams["figure.figsize"] = (9,9)
fig = plt.figure()
ax = fig.gca(projection = '3d')
ax.scatter(x1, x2, x3)
ax.set_xlabel("max power consumption", fontsize = 15)
ax.set_ylabel("CO2 emissions per hour", fontsize = 15)
ax.set_zlabel("annual energy cost", fontsize = 15, labelpad = 13)
plt.show()
```



모델을 구현하기 전에, 데이터의 분포를 확인하기 위해 독립변수를 시각화 해봤다.

```
# Normalization
min_max_scaler = MinMaxScaler()
scaled_X = min_max_scaler.fit_transform(X)
```

'연간 에너지 비용'이 다른 특성들의 범위에 비해 크다. 모든 특성들을 모두 고르게 반영하기 위해 정규화를 시행했다.

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

```
# Explore dataset
print("shape of x_train :", X_train.shape)
print("shape of y_train :", y_train.shape)
print("shape of x_test :", X_test.shape)
print("shape of y_test :", y_test.shape)
```

```
shape of x_train : (160, 3)
shape of y_train : (160,)
shape of x_test : (40, 3)
shape of y_test : (40,)
```

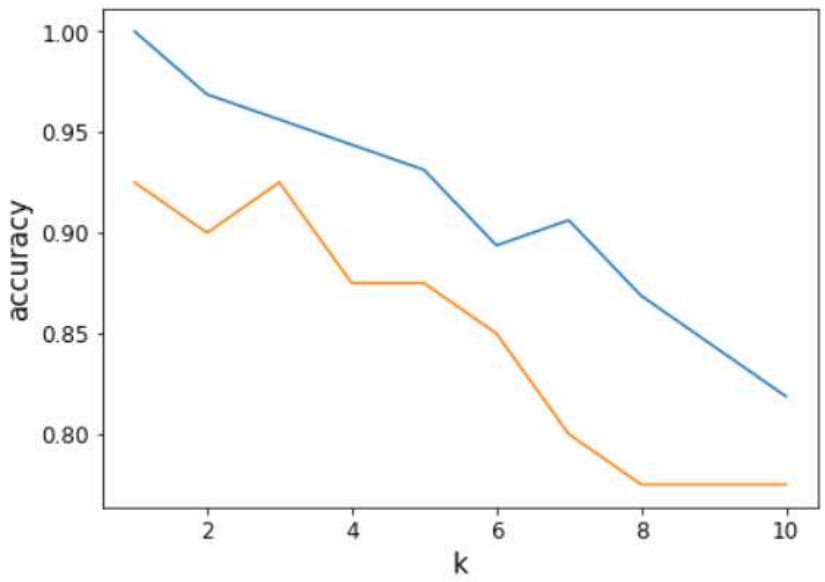
만든 데이터 셋을 훈련데이터와 테스트데이터로 나눈 후, 데이터의 개수를 확인했다.

2. 적절한 k 찾기

```
# change k
plt.rcParams["figure.figsize"] = (7,5)
train_accuracy = []
test_accuracy = []
n_neighbors = range(1, 11)

for n in n_neighbors :
    knn = KNeighborsClassifier(n_neighbors = n)
    knn.fit(X_train, y_train)
    train_accuracy.append(knn.score(X_train, y_train))
    test_accuracy.append(knn.score(X_test, y_test))

plt.plot(n_neighbors, train_accuracy, label = 'train_accuracy')
plt.plot(n_neighbors, test_accuracy, label = 'test_accuracy')
plt.xlabel("k", fontsize = 15)
plt.ylabel("accuracy", fontsize = 15)
plt.show()
```



for문을 사용하여, k에 따른 정확도를 시각화 했다. 파란색이 훈련정확도이고, 주황색이 테스트 정확도 이다.
k=1 일 때, 훈련정확도는 가장 높지만, 그에 비해서 테스트 정확도는 낮다. 오버피팅이 가장 적으면서, 테스트 정확도가 가장 높은 모델이 좋은 모델이라고 생각한다. 그렇기에 적절한 k의 값으로 3을 선택했다.

3. k-NN 구현하기

```
knn3 = KNeighborsClassifier(n_neighbors = 3)
knn3.fit(X_train, y_train)
```

k가 3인 k-NN 분류기를 구현했다.

IV. 결과 분석

1. 정확도 확인

```
print("train accuracy of knn with k=3 :", metrics.accuracy_score(y_train,  knn3.predict(X_train)))
print("test accuracy of knn with k=3 :", metrics.accuracy_score(y_test,  knn3.predict(X_test)))
```

```
train accuracy of knn with k=3 : 0.95625
test accuracy of knn with k=3 : 0.925
```

k가 3일 때, 훈련 정확도는 약 96%, 테스트 정확도는 약 93%로 계산됐다.

2. 데이터 예측하기

```
x_new = [[70, 20, 60000]]
a_pred = knn3.predict(x_new)
print("주어진 데이터의 예측 등급은",a_pred,"입니다")
```

```
주어진 데이터의 예측 등급은 [2] 입니다
```

최대소비전력량이 70, 시간당 이산화탄소 배출량이 20g,
연간 에너지 비용이 60000일 때, 에너지 소비 효율 등급을 예측한 결과이다.

V. 결론 도출

약 93%의 정확도가 나온 것으로 보아 결과는 나쁘지 않다고 생각한다. 하지만 이 분류기는 원래 다섯 개인 등급을 임의로 세 개로 나눠 학습한 모델이기 때문에, 만약 종속변수를 그대로 사용하여 학습했다면 정확도가 좀 더 떨어질지도 모르겠다.

그리고 데이터 셋을 만들 때, 전체 156377개의 데이터 중 세 요소가 모두 적절히 공개되어있는 200개의 데이터를 추출했는데, 이는 모두 품목명이 '전기냉장고'이다. 실제로 에너지소비효율등급제도는 약 30종류의 품목을 등급으로 표시하는 제도인데, 데이터가 너무 많고 빈 부분이 많아서 한 가지 품목으로만 분류기를 구현했던 점이 아쉽다.

하지만 결과를 봤을 때, 다섯 개의 등급을 종속변수로 전체 데이터를 사용하여 학습해도 좋은 모델이 만들어질 가능성은 충분히 있다고 생각한다.

VI. 참고자료

[1] [공공데이터포털] 에너지소비효율등급제도 기기부문 제품정보, https://www.data.go.kr/data/3071181/fileData.do, (2020.06.24.)