**문제 개요**

생활가전 A부서는 친환경 미니 세탁기를 개발 중에 있다. 프로젝트의 핵심은 세탁물을 자동 인식하고 이에 따른 최적 코스를 (수량, 수온, 세제종류, 세제량, 세탁시간, 세탁방법) 제시하여 세탁지수(target)를 높이는 것이다.

이를 위해 세탁지수에 영향을 주는 빨랫감의 종류와 오염량, 급수량, 세제종류, 투입량 등을 측정한 데이터를 확보하였다. 데이터의 변수명은 보안을 위해 cat0, cont0등으로 코드화 되어 있다.

아래 데이터를 사용하여 과제를 해결하고 세탁지수(target)를 예측하시오.

데이터 구성

학습데이터: train\_prob.csv, 240,000 rows, 26 columns

테스트데이터: test\_prob.csv, 60,000 rows, 25 columns

테스트정답셋: test\_prob\_ans.csv, 60,000 rows, 1 columns

|  |  |  |
| --- | --- | --- |
| 컬럼명 | 설명 | 타입 |
| Id | 실험 고유 번호 | 정수형 |
| cat0 ~ cat9 | 범주형 변수 0~9 | 문자형 |
| cont0 ~ cont13 | 연속형 변수 0~13 | 실수형 |
| target | 세탁지수 | 실수형 |

**전처리(Preprocessing)**

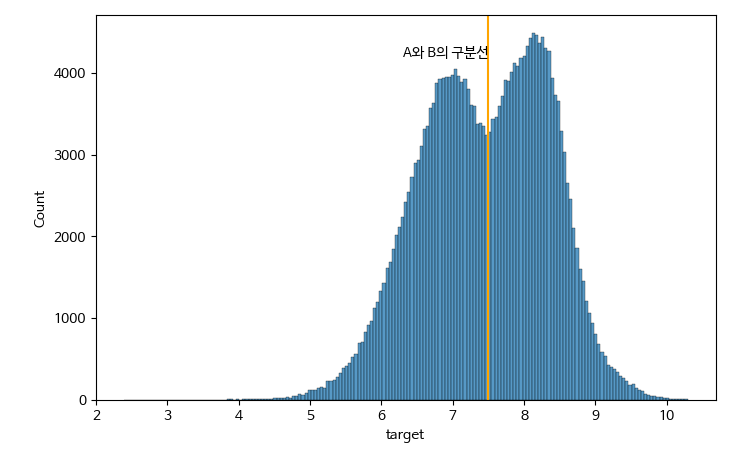
train\_prob.csv 데이터셋을 대상으로 다음 전처리를 진행하라.

Figure 1 target의 히스토그램

**단계 1**: target은 2개의 정규 분포가 합쳐진 모양을 하고 있다. 두 분포의 구분점 7.45를 기준으로 target값이 작거나 같으면 A, 초과하면 B로 나눈다. A 범주이면 True, B이면 False인 파생 변수 target A를 만든다.

**단계 2**: cat3 ~ 9 각각의 범주 중에서 빈도가 적은 범주를 target을 예측하는데 영향이 없는 방향으로 정리하려고 한다. 아래 가이드에 맞추어 cat3 ~ 9 에 대한 전처리를 수행한다.

* cat3는 B→C 로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |  |
| --- | --- | --- |
| A | C | D |
| 83,634 | 147,361 | 9,005 |

* cat4는 A→B, D→B 로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |
| --- | --- |
| B | C |
| 239,397 | 603 |

* cat6는 D→A, E→B, G→C, H→B, I→A로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |  |
| --- | --- | --- |
| A | B | C |
| 234,203 | 5,145 | 652 |

* cat7는 A→B, C→B, F→D, I→B로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |  |  |
| --- | --- | --- | --- |
| B | D | E | G |
| 4,606 | 19,784 | 214,027 | 1,583 |

* cat8는 B→G, F→E로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | C | D | E | G |
| 30,338 | 96,743 | 2,953 | 76,085 | 33,881 |

* cat9는 C→H, D→B, E→L로 치환한다.

Hint] 치환 후의 빈도 테이블

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | F | G | H | I | J | K | L | M | N | O |
| 10,678 | 2,846 | 85,944 | 8,320 | 19,987 | 40,070 | 5,501 | 16,743 | 33,793 | 7,819 | 3,331 | 4,968 |

이 과정을 거친 후 데이터셋을 (basetable)이라고 한다.

**[문제1]**

범주형 변수의 조합의 특성을 파악하고자 한다. 다음 과정을 수행하라.

**단계 1-1**: basetable을 복사하여 prob1을 만들고 cat0 ~ 9 변수 순서대로 문자열 결합하여, 10자리 문자형 파생 변수(cat\_c)를 생성하라.

**단계 1-2**: 50개 이상 행으로 이루지고, target에 대한 정규성을 부정할 수 없는 cat\_c를 찾아 cat\_c\_A와 cat\_c\_B를 구하라. 정규성 검정은 Kolmogolov-Smironov 검정을 사용한다. (유의 수준: 5%)

- cat\_c의 행의 개수가 많은 순으로 cat\_c 3개를 cat\_c\_A

- cat\_c의 행의 개수가 적은 순으로 cat\_c 3개를 cat\_c\_B

\* 행의 개수가 같을 경우에는 p-value가 높은 순으로 고른다.

**함수 가이드**

scipy.stats 제공 기능 활용

문제 지시사항 외 Default 값 사용

**단계 1-3**: bartlett 검정을 사용하여 cat\_c\_A에 해당하는 범주간에 target의 등분산성을 조사했을 때 검정 통계량을 A라 한다. 그리고 bartlett 검정을 사용하여 cat\_c\_B에 해당하는 범주간에 target의 등분산성을 조사했을 때 검정 통계량을 B라고 한다.

**함수 가이드**

scipy.stats 제공 기능 활용

문제 지시사항 외 Default 값 사용

**단계 1-4**: 등분산성을 부정할 수 없는 경우는 경우는 단 하나 있다. 이 경우에 해당하는 행들을 대상으로 해당 범주 3개에 대한 분산분석(ANOVA)를 수행하고 p-value를 C라고 한다.

**함수 가이드**

scipy.stats 제공 기능 활용

문제 지시사항 외 Default 값 사용

**A + B + C 를 소수점 넷째 자리에서 반올림하여 셋째 자리까지 표시하라.**

Ex) 123.456

**[문제2]**

연속형 변수 cont0 ~ 13 중에서, 서로 피어슨 상관도(Pearson correlation)가 높은 변수들 찾고, 이들을 PCA를 통한 차원 축소를 하여 성능 향상이 있는지 확인하고자 한다.

**단계 2-1**: basetable 에서 인덱스와 컬럼 모두 cont0 ~ 13이고, 인덱스와 컬럼에 해당하는 변수 간에 피어슨 상관도(Pearson correlation)를 값으로 하는 대각 대칭형인 데이터프레임을 corr을 만든다.

**단계 2-2**: 계층적 군집화를 활용하여 다음 개념도와 같이 연관성이 높은 변수 군집을 찾아 보고자 한다. 초기에는 cont0~13 변수 각각을 하나의 군집으로 배정한 상태에서 시작하여, 군집 간 결합 조건으로 두 군집에 포함된 변수간의 피어슨 상관도 절대값의 평균이 0.5이상 일 때, 해당 수치가 높은 것을 우선으로 군집들을 결합한다. 군집의 크기가 가장 큰 변수들을 구하라.

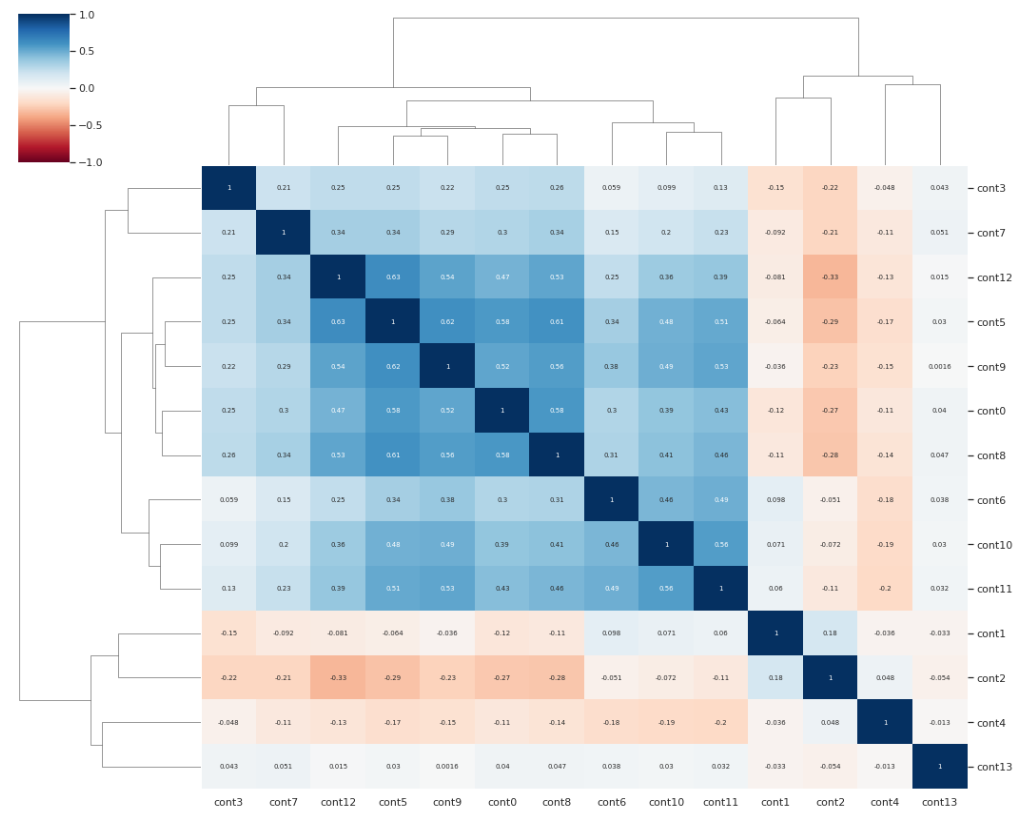


Figure 2 변수 군집화 개념도 (실제 결과와는 상관 없음)

Hint] cont0 ~ 13간의 거리는 [단계 2-1] corr을 이용하여 군집 결합 조건을 고려하여 구한다. (두 변수간에 피어슨 상관도의 절대값이 높을 수록 거리는 가깝다.)

**함수 가이드**

sklearn.cluster.AgglomerativeClustering

문제 지시사항 외 Default 값 사용

**단계 2-3**: basetable에서 cat0 ~ 9을 각 변수의 첫 번째 범주(알파벳 순)를 제외하고 원핫 인코딩(One-hot encoding)하여 prob2를 만든다.

**단계 2-4:** prob2에서 80%는 학습용 prob2\_train으로 20%는 테스트용 prob2\_test으로 분리한다.

**함수 가이드**

sklearn.model\_selection.train\_test\_split, random\_state=123

문제 지시사항 외 Default 값 사용

**단계 2-5:** prob2\_train으로 [단계 2-2]에서 도출된 변수들로 PCA 모델을 만들고 분산 설명률이 높은 순으로 누적 분산 설명률이 0.8을 넘기는 최소의 컴포넌트를 prob2\_train에 추가 한다. 동일한 PCA 모델로 prob2\_test를 변환하여 같은 수의 컴포넌트를 prob2\_test에 추가 한다.

**함수 가이드**

sklearn.decomposition.PCA (입력의 순서는 cont 뒤의 번호가 낮은 것에서 높은 것 순으로 한다, random\_state=123)

문제 지시사항 외 Default 값 사용

**단계 2-6:** prob2\_train에서 target을 선형회귀모델로 학습하고 prob2\_test를 다음 성능 지표로 평가한 값을 A라고 한다.

입력변수: cont0 ~ 13, cat0 ~ 9 원핫-인코딩 후 나온 파생 변수 (변수명, 범주명 알파벳 순서)

대상변수: target

성능지표:



Hint] 학습데이터의 형태는 (192,000, 43) 이다.

**함수 가이드**

sklearn.linear\_model.LinearRegression

문제 지시사항 외 Default 값 사용

**단계 2-7:** prob2\_train에서 cont0 ~ 13 중에서 PCA에서 사용했던 변수를 제외하고 단계 2-5에서 추가한 PCA 컴포넌트를 포함하여 target로 선형회귀모델을 학습한다. prob2\_test로 동일한 지표로 대한 성능 평가한 값을 B라 한다.

입력변수: cont0 ~ 13중 PCA에 사용한 것을 제외한 나머지 변수, PCA를 통한 파생 변수를 분산 설명력이 높은 순으로, cat0 ~ 9까지 원핫-인코딩 후 나온 파생 변수 (변수명, 범주명 알파벳 순서)

대상변수: target

성능지표:



**함수 가이드**

sklearn.linear\_model.LinearRegression

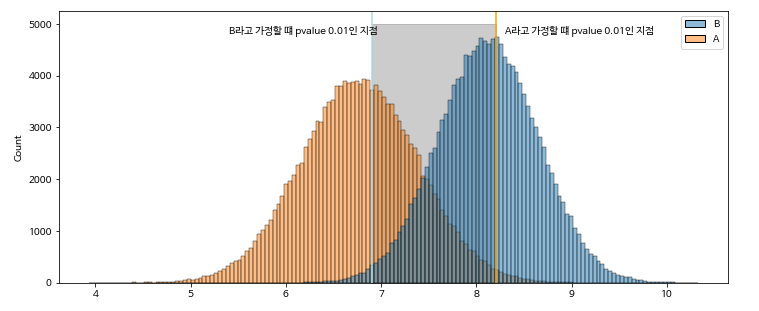
문제 지시사항 외 Default 값 사용

**A-B 를 소수점 넷째 자리에서 반올림하여 셋째 자리까지 구하라.**

ex) 123.456

**[문제3]**

전처리 단계에서 target이 두 개의 정규 분포의 혼합한 형태 보이고 있음 확인했고, target을 7.45 경계값을 기준으로 A와 B범주로 나누었다. Gaussian Mixture라는 모델을 사용하여 두 개 정규 분포의 매개 변수를 측정했고, target의 분포를 시각화했다. 아래 시각화 그래프에서 A, B가 혼재하여 A 또는 B라고 확신할 수는 없는 구간이 존재한다. 확률을 기반으로, A 또는 B라고 확신할 수 있는 관측치만을 가지고 A / B를 분류 모델을 만들고자 한다. 다음 단계들을 수행하라.



**단계 3-1:** A 범주의 평균은 6.769 표준편차는 0.616이고 범주 B일 때 평균은 8.123 표준편차는 0.527이다. 측정한 값을 범주A, B의 모집단의 target의 평균과 표준편차라고 가정한다. 두 개의 정규 분포를 이용하여, 귀무가설이 ‘target은 A 범주에 해당한다’일 때 p-value를 구하여 파생변수(prob\_A)를 만들다. 반대로, 귀무가설이 ‘target은 B 범주에 해당한다’일 때 p-value를 구하여 파생변수(prob\_B)를 만든다. basetable에 두 개의 파생 변수 prob\_A, prob\_B를 추가하여 prob3 데이터프레임을 만든다.

Hint] scipy.stats.norm

**단계 3-2:** prob3 중에서 target이 A 또는 B라고 신뢰할 수 있는 행들만을 골라내고자 한다. prob\_B가 0.01 미만이거나, prob\_A가 0.01 미만인 것만 남긴다. (즉, 위 그림에서 회색 영역에 해당하는 부분은 제외)

Hint] 행의 개수는 124,600가 된다.

**단계 3-3:** prob3에 cat0 ~ 9를 제외하는 범주 없이 원핫 인코딩(One-Hot Encoding) 한다.

**단계 3-4:** prob3를 학습용 80%, 그 외 20%는 테스트용으로 나누어 prob3\_train, prob3\_test로 한다.

**함수 가이드**

sklearn.model\_selection.train\_test\_split, random\_state=123

문제 지시사항 외 Default 값 사용

**단계 3-5:** prob3\_train으로 target\_A를 예측하는 xgboost 분류 모델을 다음과 같은 설정으로 prob3\_train를 사용하여 학습한다. prob3\_test으로 다음 성능지표로 평가한 값을 C 라고 한다.

입력변수: cont0 ~ 13, cat0 ~ 9까지 원핫-인코딩 파생 변수 (변수명, 범주순으로 알파벳 순서)

대상변수: targetA

성능지표:



하이퍼 파라미터(Hyper-Parameter, 초매개변수) 설정:

**함수 가이드**

xgboost 제공 기능 활용, 문제 지시사항 외 Default 값 사용

|  |  |
| --- | --- |
| 항목 | 값 |
| 트리의 최대 깊이 | 2 |
| L1 규제 | 0.1 |
| L2 규제 | 0.1 |
| 트리 당 컬럼 샘플링 비율 | 0.25 |
| 트리의 수 | 500 |
| random\_state | 123 |

**C 를 소수점 넷째 자리에서 반올림하여 셋째 자리까지 출력하라.**

ex) 3.142

**[문제 4]**

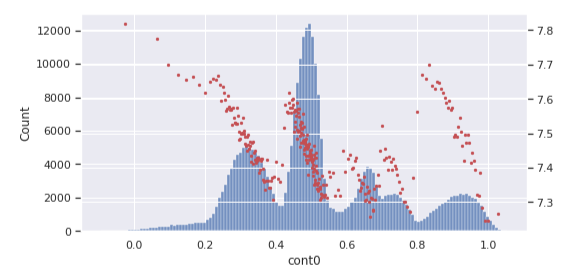


Figure 3 cont0의 빈도그래프와 cont0 분위수(300분위)별 target평균 산포도 그래프

위 차트는 cont0 의 히스토그램과 cont0를 300개의 분위 구간으로 나누어 x값은 분위 구간의 중간값, y값은 분위 구간에서 target의 평균을 산포도 그래프로 동시에 출력했다. 좌측 y축은 히스토그램의 빈도수이고, 우측 y축은 target의 평균이다.

히스토그램에서 동일한 봉우리 구간 내에서 cont0와 target간에 선형적 관계가 있음이 확인되었다. 이 점을 이용하여 파생 변수를 만들어서 회귀모델을 만들어 간다.

**단계 4-1:** basetable에서 cat0 ~ 9을 첫 번째 범주(알파벳 순)는 제외하고 원핫 인코딩(One-Hot Encoding)하여 prob4데이터셋을 만든다.

**단계 4-2:** prob4의 80%를 prob4\_train으로 20% prob4\_test로 나눈다.

**함수 가이드**

sklearn.model\_selection.train\_test\_split, random\_state=123

문제 지시사항 외 Default 값 사용

**단계 4-3:** prob4\_train에서 cont0를 100개의 우측이 닫힌 (ex. i번째 분위수 < cont0 ≤ i+1 번째 분위수) 분위 구간으로 나누고 각각의 분위 구간에서 target의 평균을 구해 파생 변수 cont0\_q를 만든다. prob4\_test에는 prob4\_train을 나누었던 구간과 평균으로 cont0\_q를 만든다. 이 때, prob4\_test의 cont0값이 train의 첫 번째 구간보다 작다면, 첫 번째 구간으로 간주하고, 마지막 구간보가 크다면, 마지막 구간으로 간주한다.

Hint 1] pandas.Series.quantile, pandas.cut 활용을 권장한다.

Hint 2] 두 번째 분위 구간은 0.111 < cont0 <= 0.176이고 target의 평균은 7.672 (셋째 자리까지 반올림)

Hint 3] cont0\_q의 평균은 7.46, 표준 편차는 0.10 (소수점 둘째 자리까지 반올림)

**단계 4-4**: 나머지 변수 cont1~13에 대해서도 해당 작업을 진행한다. 생성한 파생 변수명은 원래의 변수명에 \_q를 결합한다. (ex. cont1 -> cont1\_q)

**단계 4-5:** prob4\_train에서 target을 대상으로 선형회귀모델을 학습하고, prob4\_test를 다음 지표로 성능 평가한 값을 D라고 한다. 선형회귀 모델은 아래와 같은 손실함수로 학습을 한다.

, **α = 0.1**

Hint] sklearn.linear\_regression.Ridge 또는 Lasso 중 요건에 맞는 것을 사용한다.

입력 변수: cont0\_q ~ cont13\_q, cat0 ~ 9까지 원핫-인코딩 후 나온 파생 변수 (변수명 다음 범주명으로 알파벳 순서이다)

대상 변수: target

성능 지표:



**D를 소수점 셋째 자리에서 반올림하여 둘째 자리까지 출력하라.**

ex) 123.45

**단계 4-6 [Optional]**: 최적의 구간 수를 교차 검증을 이용하여 구하고자 한다. (100, 300, 500) 중에서 평균제곱오차(MSE, mean squared error)의 평균이 가장 낮은 구간 수를 E로 한다.

요건] sklearn.model\_selection.KFold n\_splits=5, random\_state=123, shuffle=True

**[문제 5]**

XGBoost 분류 모델의 최적의 하이퍼 파라미터(Hyper-Parameter, 초매개변수)를 탐색한다.

**단계 5-1:** basetable에 cat0 ~ 9 제외하는 범주 없이 원핫 인코딩(One-Hot Encoding) 하여 prob5를 만든다.

**단계 5-2:** prob5를 80%는 학습용 prob5\_train으로, 20%는 테스트용 prob5\_test로 나눈다.

**함수 가이드**

sklearn.model\_selection.train\_test\_split, random\_state=123

문제 지시사항 외 Default 값 사용

**단계 5-3:** 다음 회귀 분석 모델을 아래 3개의 하이퍼 파라미터(Hyper-Parameter, 초매개변수)를 사용하여 prob5\_train으로 target를 학습시킨다. prob5\_test으로 테스트했을 때 최적의 성능을 낸 하이퍼 파라미터 셋의 번호를 A, 성능지표를 B라고 한다.

입력 변수] cont0~13, cat0 ~ 9을 원핫 인코딩하여 나온 파생 변수(변수명, 범주명 알파벳순)

대상 변수] target

성능 지표]



|  |  |
| --- | --- |
| **No** | **하이퍼 파라미터(Hyper-parameter, 초매개변수)** |
| 1 | colsample\_bytree=0.25, n\_estimators=300 |
| 2 | colsample\_bytree=0.5, n\_estimators=300 |
| 3 | colsample\_bytree=0.25, n\_estimators=500 |

공통적으로 트리의 최대 깊이는 2, 랜덤 시드는 123이다.

**함수 가이드**

xgboost 제공 기능 활용, 문제 지시사항 외 Default 값 사용

A + B 를 소수점 셋째 자리에서 반올림하여 둘째 자리까지 구하라.

**문제 6**

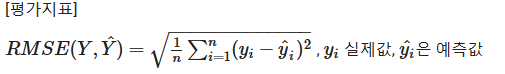
[Kaggle 형] train\_prob.csv로 문제 target을 예측하는 모델을 만들고, test\_prob.csv에 대한 target 예측하여 다음과 같은 형식의 answer6.csv를 만들어라.

id, target

0, 6.9

5, 7.8

...



모범 답안

|  |  |  |
| --- | --- | --- |
| 문제 | 시험장 환경 | BDC 환경 |
| 1 | 10.740 | 10.740 |
| 2 | -0.004 | -0..004 |
| 3 | 0.655 | 0.658 |
| 4 | 0.72 | 0.72 |
| 5 | 3.72 | 3.71 |
| 6 | 0.85 근처 | 0.85 근처 |