**문제 개요**

다음은 폴더블 폰의 힌지에 들어가는 스프링 내구력을 테스트한 실험 결과이다. 스프링 측정값과 스프링에 가한 부하 정보와 함께, 테스트 통과/실패 (failure) 결과가 기재되어 있다. 개발부서는 테스트 비용을 줄이기 위해 failure 여부를 맞추는 모델을 만들고자 한다.

변수명은 보안을 위해 measurement\_0과 같이 익명화되었다.

데이터 구성

학습데이터: train\_prob.csv, 21,458 rows, 25 columns

테스트데이터: test\_prob.csv, 5,112 rows, 24 columns,

테스트정답셋: test\_prob\_ans.csv, 5,112 rows, 1 columns

|  |  |  |
| --- | --- | --- |
| 컬럼명 | 설명 | 타입 |
| id | 실험 고유 번호 | 정수형 |
| product\_code | 스프링 코드 | 범주형 |
| loading | 스프링에 가한 부하 | 실수형 |
| attribute\_0 | 구성 소재1 | 범주형 |
| attribute\_1 | 구성 소재2 | 범주형 |
| attribute\_2 | 구성 소재3 | 정수형 |
| attribute\_3 | 구성 소재4 | 정수형 |
| measurement\_0 ~ 17 | 측정값 0~17 | 실수형 |
| failure | 성공여부 | 이진형(0, 1) |

**전처리(Preprocessing)**

train\_prob.csv를 불러 온다. 이를 basetable이리고 한다.

**단계 1**: basetable에 measurement\_3 ~17 각각의 행이 결측인지 나타내는 파생 변수를 만든다. 파생 변수는 이진 형식이고, False는 미결측 True는 결측을 의미한다. 파생 변수의 이름은 measurement 번호에 따라 isna\_3 ~ 17로 한다.

**단계 2**: 이 과제를 맡은 데이터분석가 지희는 measurement\_3~17의 결측치 처리 방안을 고민하던 중, 개발부서에서 measurement\_17은 product\_code별로 failure를 예측하기 위해 measurement\_3 ~ measurement\_9을 다음과 같이 선형 조합하여 생성한 값이라는 정보를 받았다.

이는 즉,

….

와 같이 measurement\_3 ~ measurement\_9의 각 변수들도 나머지 변수들과 선형 관계를 지닌다. 이 점을 이용하여 대상 변수를 번갈아 가면서 예측 모델을 만들어 최대한 원래 값에 가깝게 복원할 수 있다. 이러한 반복적인 결측치 복원 방법을 사내 데이터분석 연구소에 문의 했더니 다음과 같은 가이드를 주었다.

|  |
| --- |
| sklearn 모듈에 아직은 실험 단계이지만, 비슷한 경우에 문제 없이 사용했던 사례가 있어 의견을 드립니다.  from sklearn.experimental import enable\_iterative\_imputer 구문을 사용하여 실험 단계인 모듈을 활성화하고, sklearn.impute.IterativeImputer를 사용한다면 원하는 결과를 얻을 수 있습니다. |

가이드의 내용을 참조하여 basetable의 measurement\_3~9와 measurement\_17 결측치를 복원하라.

입력 변수] measurement\_3 ~ 9, measurement\_17 (입력 변수 순서에 유의)

**함수 가이드**

sklearn.experimental.enable\_iterative\_imputer

sklearn.impute.IterativeImputer, random\_state=123

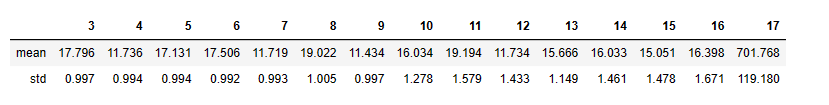
sklearn.linear\_model.LinearRegression

문제 지시사항 외 Default 값 사용

**단계 3**: measurement\_10~16까지의 결측치는 모두 product\_code별 평균으로 대치한다.

Hint] 전처리 단계에서 보간 결과를 확인해 보기 위한 각 변수의 평균과 표본표준편차.

열의 이름의 숫자는 measurement\_ 번호, 값은 소수점 3째 자리까지 반올림

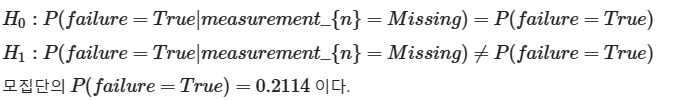


※ 시험장 환경에서의 수행 결과를 바탕으로 만들어진 테이블입니다. 실습환경에 따라 차이가 날수 있으니 유의하세요.

**[문제1]**

(basetable을 사용) measurement\_3~16까지 결측 여부가 failure에 영향이 있는지를 파악하고, failure를 분류하는 데 도움이 될 만한 것은 예측 모델의 입력 변수로 사용하고자 한다. 이를 위해 전처리 과정에서 뽑아낸 isna\_3~16을 활용한다.

n이 3부터 16까지, 즉 measurement\_3~16까지 다음의 검정을 수행한다.



**단계 1-1**: 우선, measurement\_3으로 위 검정을 시행해보자.



으로 바꿀 수 있다.

 은 표본수가 충분하여 중심극한정리에 의해 정규분포를 따르는 것은 분석가 간에 이견이 없다고 한다. 위 검정의 p-value를 구하여 보고 힌트에 주어진 p-value와 비교하여 검정 방법에 문제가 없음을 확인하라.

**함수 가이드**

scipy.stats 제공 기능 활용, 문제 지시사항 외 Default 값 사용

Hint: p-value는 0.0037(소수점 다섯째 자리에서 반올림하여 넷째 자리까지 표시)

**단계 1-2**: measuremenet\_3을 포함하여 measurement\_4 ~ 16까지 위 검정을 반복하고 귀무가설을 기각할 수 있는 경우의 p-value의 합을 A라고 한다. (유의 수준은 5%로 한다.)

**단계 1-3:** 검정 결과 귀무가설을 기각할 수 있는 경우는 총 두 건이다. 해당 파생 변수명의 뒷 자리 번호 순으로 na\_1, na\_2로 파생 변수를 만들어 prob1 데이터셋을 생성하라.

**A의 값을 소수점 넷째 자리에서 반올림하여 셋째 자리까지 출력하시오.**

Ex) 0.052

**단계 1-4 (Optional):** measurement\_3~measurement\_16까지, 각 행의 결측치 수가 failure와 연관성이 있을지 검정하시오.

Hint] scipy.stats.chi2\_contingency

**[문제2]**

첫째는 스프링 개발 업체들은 실험이 제품 별로 공정하게 진행이 됐는지를 의문을 가지고 있다. product\_code에 따라 개발 업체가 다르다. product\_code에 대해서 스프링에 가한 부하(loading)를 동일하게 했는지 조사하라.

둘째는, attribute\_0와 attribute\_1은 스프링을 구성하는 주요 소재이다. failure와는 관계가 없음이 이전에 검증되었다. 하지만, 이에 대한 재확인 요청을 받아 attribute\_0와 attribute\_1은 failure와 상관없음을 확인한다.

이를 위해 다음 단계를 수행하라.

**단계 2-1:** prob1에서 입력 변수 loading에 결측이 없는 행들을 뽑아 prob2 데이터프레임을 만든다.

Hint] prob2의 데이터 수는 21,257 이다.

**단계 2-2**: prob\_2에 loading의 각 행들에 자연 로그 함수를 적용하여 파생 변수 loading\_log를 만든다.

**단계 2-3:** loading\_log가 product\_code에 대해 정규성을 지니고 있는지 확인하고자 한다. 이를 위해 Jarque-Bera로 검정하고 결과의 p-value가 0.05가 넘는 product\_code의 수를 B라고 한다.

**함수 가이드**

scipy.stats 제공 기능 활용, 문제 지시사항 외 Default 값 사용

**단계 2-4:** loading\_log 변수를 product\_code로 구분했을 때, 등분산성을 보이는지 Bartlett 검정을 통해 확인한다. 검정 결과에서 p-value를 C라고 한다.

**함수 가이드**

scipy.stats 제공 기능 활용, 문제 지시사항 외 Default 값 사용

**단계 2-5:** product\_code에 대한 분산분석(ANOVA)을 통해서 loading\_log 평균에 차이가 있는지 검정한다. 그 결과 중 p-value를 D라고 한다.

**함수 가이드**

scipy.stats 제공 기능 활용, 문제 지시사항 외 Default 값 사용

**단계 2-6:** Chi-square 검정을 통해 attribute\_0, attribute\_1의 결합이 failure와 연관이 있는지 조사하라. (유의 수준 1%) 연관이 있다면 E값은 1 없으면 0으로 한다.

attribute\_0=material\_7, attribute\_1=material\_8 이라면, 이 둘의 결합은matertial\_7material\_8를 의미한다.

**함수 가이드**

scipy.stats.chi2\_contingency, correction=False, 문제 지시사항 외 Default 값 사용

**B + C + D + E의 값을 소수점 셋째 자리에서 반올림하여 둘째 자리까지 출력하시오.**

Ex) 3.16

**[문제3]**

로지스틱 회귀모델로 수치형 변수measurement\_0 ~ 17, loading과 이진형인 na\_1, na\_2 중에서 최적의 성능을 보이는 입력 변수들을 찾고자 한다.

**단계 3-1:** prob1을 복사하여 prob3을 만든다. loading의 결측치는 loading의 평균으로 대치한다.

**단계 3-2:** prob3를 80%는 학습데이터 prob3\_train으로 20%는 테스트데이터 prob3\_test로 나눈다. prob3\_train의 failure가 1인 비율과 prob3\_test의 failure가 1의 비율을 동일하게 한다.

**함수 가이드**

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

train과 test의 failure의 비율은 stratify 매개 변수를 이용하여 맞춘다.

문제 지시사항 외 Default 값 사용

**단계 3-3**: prob3\_train의 수치형 입력 변수loading, measurement\_0 ~ 17을 표준화한다. prob3\_train의 표준화 설정으로 prob3\_test의 loading, measurement\_0 ~ 17에도 적용한다. 표준화 처리한 prob3\_train과 prob3\_test는 문제4와 문제 5에서 사용한다.

**함수 가이드**

sklearn.preprocessing 제공 기능 활용, 문제 지시사항 외 Default 값 사용

**단계 3-4:** 로지스틱 회귀모델을 사용하여 loading, measurement\_0~17과 na\_1, na\_2를 입력 변수로 하여 prob3\_train을 학습한다. 로지스틱 회귀모델을 prob3\_test로 성능을 측정한 값을 A라고 한다.

입력 변수: loading, measurement\_0~17, na\_1, na\_2

대상 변수: failure

성능 지표: AUC(area under of ROC curve)

**함수 가이드**

sklearn.linear\_model.LogisticRegression, solver='lbfgs'

sklearn.metrics.roc\_auc\_score

문제 지시사항 외 Default 값 사용

**단계 3-5:** loading, measurement\_0 ~ 17, na\_1, na\_2를 후보 입력 변수로 한다. 전진 선택법을 사용하여 이 후보 입력 변수 중에서 최적의 성능을 보이는 입력 변수의 조합을 찾는다. 전진 선택법의 선택 기준은 prob3\_train을 대상으로 5겹 층화교차검증(5-Fold stratified cross validation)을 하고 겹외(OOF, Out-Of Fold) 성능의 평균값으로 한다. 전진 선택 과정에서 선택했던 변수를 제외하지 않는다.

입력 변수: 본 단계 요건 참고

대상 변수: failure

성능 지표: AUC(area under of ROC curve)

**함수 가이드**

mlxtend.feature\_selection.SequentialFeatureSelector

sklearn.linear\_model.LogisticRegression, solver='lbfgs'

sklearn.model\_selection.StratifiedKFold, random\_state=123, shuffle=True

문제 지시사항 외 Default 값 사용

**단계 3-6:** 단계 3-5에서 찾은 최적의 입력 변수 조합으로 로지스틱 회귀모델을 사용하여 prob3\_train을 학습하고 prob3\_test로 성능을 측정한 값을 B라고 한다.

입력 변수: **단계 3-5**에서 도출한 최적의 입력 변수 조합

대상 변수: failure

성능 지표: AUC(area under of ROC curve)

**함수 가이드**

sklearn.linear\_model.LogisticRegression, solver='lbfgs'

sklearn.metrics.roc\_auc\_score

문제 지시사항 외 Default 값 사용

**A-B값을 소수점 넷째 자리에서 반올림하여 셋째 자리까지 출력하시오**

**Ex) 1.135**

**[문제 4]**

차원 축소 기법을 통한 데이터의 특성과 failure 분류 성능을 높힐 만한 요소를 살펴 본다.

첫째로, loading을 제외하고, measurement\_0 ~ 17을 입력으로 failure를 대상 변수로 Linear Discrimant Analysis(LDA) 모델을 만든다. 범주가 두 개인 failure를 분류한다는 점에서 LDA 모델은 measurement\_0 ~ 17를 한 개의 경계점으로 failure를 최대한 정확하게 구분하도록 하나의 연속형 변수로 변환한다. 실험 대상의 내구력을 나타낸다고 할 수 있는LDA 변환값과 실험에서 스프링에 가한 부하(loading)와 상관도를 측정하여, 스프링에 따라 부하(loading)를 조정한 정도를 살펴본다..

둘째로, PCA를 사용하여 차원 감소로 failure 분류 성능에 얼마나 효과가 있을지 살펴본다.

문제3에서 사용했던, 전처리(loading 결측치 처리와 표준화 과정을 거친) 과정을 거친 prob3\_train과 prob3\_test를 사용한다.

**단계 4-1:** prob3\_train에서 measurement\_0 ~ 17을 입력으로 failure를 대상 변수로 하여 LDA(Linear Discriminant Analysis) 모델을 학습한다. measurement\_0 ~ 17에 대한 LDA의 변환값과 loading과 스피어만 상관도 (spearman correlation)의 p-value를 구하여 A라고 한다.

입력 변수] measurement\_0 ~ 17 (순서에 유의 하시오)

대상 변수] failure

**함수 가이드**

sklearn.discriminant\_analysis 제공 기능 활용

scipy.stats.spearmanr

문제 지시사항 외 Default 값 사용

**단계 4-2:** prob3\_train에서 measurement\_0 ~ 17을 대상으로 주성분분석(Principal Component Analysis, PCA) 모델을 학습한다. 분산 설명율이 높은 순으로 주성분을 변수명을 pca\_0 ~ 17하여 prob3\_train에 추가하여 prob4\_train을 만든다. prob3\_test에 prob3\_train를 학습했던 PCA 모델로 동일한 방법으로 pca0 ~17 파생 변수를 추가하여 prob4\_test를 만든다.

입력 변수] measurement\_0 ~ 17 (순서에 유의 하시오)

**함수 가이드**

sklearn.decomposition.PCA, random\_state=123

문제 지시사항 외 Default 값 사용

**단계 4-3:** 초기에 loading을 입력 변수로 하여 prob4\_train을 학습하고, prob4\_test에 대한 성능을 측정한다. 여기에 pca\_0에서 pca\_17까지 입력 변수를 하나씩 추가 하면서, 즉 분산 설명율이 높은 순으로 컴포넌트를 하나씩 추가하여 prob4\_train를 학습하고 prob4\_test의 성능을 측정 했을 때, 최적의 성능을 보인 컴포넌트들의 분산 설명율의 합을 B라고 한다. (만일 없다면 B = 0이다.)

입력 변수: 설명 참고

대상 변수: failure

성능 지표: AUC(area under of ROC curve)

**함수 가이드**

sklearn.linear\_model.LogisticRegression, solver=’lbfgs’

문제 지시사항 외 Default 값 사용

**A + B를 소수점 셋째 자리에서 반올림하여 둘째 자리까지 구하라.**

Ex) 12.34

**[문제 5]**

랜덤포레스트 분류기(Random-Forest Classifier)의 최적의 하이퍼 파라미터(Hyper-Parameter, 초매개변수)를 탐색하고자 한다.

문제3에서 사용했던, 전처리(loading 결측치 처리와 표준화 과정을 거친) 과정을 거친 prob3\_train과 prob3\_test를 사용한다.

**단계 5-1**: sklearn에서 제공하는 랜덤포레스트 분류기(Random-Forest Classifier)의 하이퍼 파라미터 중 n\_estimators, max\_depth 그리고 min\_samples\_split의 최적 조합을 탐색한다. 탐색 값은 아래에 제공한 하이퍼 파라미터의 모든 조합이다. prob3\_train을 대상으로 5-겹 층화교차검증(5-fold stratified cross validation)으로 각각 층의 겹외셋(OOF set, Out-Of-Fold set)의 성능 평균을 기준으로 하이퍼 파라미터를 선택한다.

- n\_estimators: [5, 10, 15]

- max\_depth: [5, 6, 7]

- min\_samples\_split: [256, 512]

Hint] 모든 하이퍼 파라미터의 조합의 수는 18개이다

입력 변수: loading, measurement\_0 ~ 17, na\_1, na\_2 (순서에 유의)

대상 변수: failure

성능 지표: AUC (area under of ROC curve)

**함수 가이드**

sklearn.ensemble.RandomForestClassifier, random\_state=123

sklearn.model\_selection.StratifiedKFold, random\_state=123, shuffle=True

itertools.product 필요시 사용

sklearn.model\_selection.cross\_val\_score 필요시 사용

sklearn.model\_selection.GridSearchCV 필요시 사용

문제 지시사항 외 Default 값 사용

**단계5-2**: 단계 5-1에서 구한 최적 하이퍼 파라미터로 설정한 랜덤포레스트 분류기(Random-Forest Classifier)를 사용하여 prob3\_train 학습하고, prob3\_test로 성능을 측정하여 이 값을 A라고 한다.

입력 변수: loading, measurement\_0 ~ 17, na\_1, na\_2 (순서에 유의)

대상 변수: failure

성능 지표: AUC (area under of ROC curve)

**함수 가이드**

sklearn.ensemble.RandomForestClassifier, random\_state=123

문제 지시사항 외 Default 값 사용

**A값을 소수점 넷째 자리에서 반올림하여 3째 자리까지 출력하시오**.

Ex) 0.123

**문제 6**

[Kaggle 형] train\_prob.csv로 문제 failure 예측하는 모델을 만들고, test\_prob.csv에 대한 failure가 1일 확률 예측하여 다음과 같은 형식의 answer6.csv를 만들어라. 측정 지표는 AUC(area under of ROC curve)이다. id 는 테스트 케이스의 id 이고, failure에는 failure가 1이 될 확률이다.

id,failure

16115, 0.1

16116, 0.2

…

모범 답안

|  |  |  |
| --- | --- | --- |
| 문제 | 시험장 환경 | BDC 환경 |
| 1 | 0.030 | 0.030 |
| 2 | 4.36 | 4.36 |
| 3 | -0.005 | -0.003 |
| 4 | 1.19 | 1.19 |
| 5 | 0.569 | 0.573 |
| 6 | 0.58~0.59 | 0.58~0.59 |