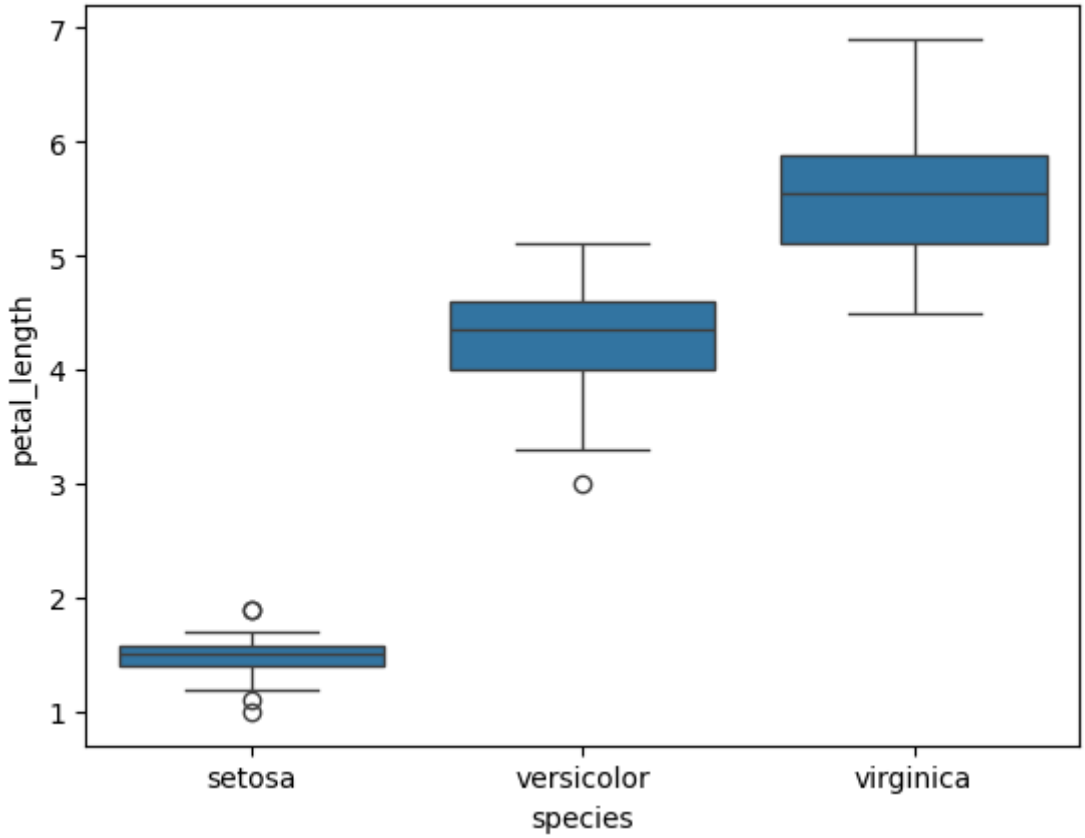
기초통계, ML 과제

27기 김현수

[기초통계]

1. Iris 데이터를 불러와서 확인한 결과, 150개의 행과 5개의 열로 구조로 되어있는 것을 확인하였다.
   추가로, iris.head()를 시행하여 실제 iris 데이터가 어떻게 구성되어있는지 확인하였다.
2. 기술통계량을 산출하였다.
   각 항목별 data 개수, 평균, 표준편차, 최솟값, 사분위수, 최댓값을 모두 확인하였다.
3. boxplot으로 각 species별 petal_length의 분포를 나타내었다.



   virginica, versicolor, setosa 순으로 평균이 높고, petal_length의 분포 또한 비슷한 순으로 나타난다. species에 따라 petal length가 크게 차이나는 것을 확인할 수 있다.
4. 정규성 검정을 위해, setosa, versicolor, virginica의 petal length의 데이터로 Shapiro-wilk 검정을 하였고, p-value가 각각 0.0548, 0.1584, 0.1097로 모두 귀무가설을 기각할 수 없었고, 따라서 정규분포를 따른다고 볼 수 있다.

5. 등분산성 검정을 위해, 세 species의 petal length 데이터로 levene 검정을 하였고, p-value가 매우 작으므로 등분산성을 따른다고 볼 수 없다.
6. ANOVA 검정을 하기 위해, 귀무가설을 세웠다.
   Null Hypothesis: 3개 Species간 petal length의 평균의 차이가 없다.
   Alternative Hypothesis: 3개 Species간 petal length 평균의 차이가 있다.
7. ANOVA 실행 결과, 다음과 같다.

	df	sum_sq	mean_sq	F	PR(>F)
C(species)	2.0	437.1028	218.551400	1180.161182	2.856777e-91
Residual	147.0	27.2226	0.185188	NaN	NaN

   ANOVA 검정의 결과는 위와 같다.
   p-value가 0.05보다 작기 때문에, 귀무가설을 기각한다.
8. 6에서 세운 귀무가설이 기각되었기 때문에, 모든 species간 평균의 차이가 있는지 확인하기 위해 사후검정을 진행하였다.

   Setosa와 versicolor, setosa와 virginica, versicolor와 virginica 모두 reject 가 True로 나왔기 때문에, 평균이 같다는 가설을 기각한다고 볼 수 있고, 두 species 사이의 평균이 유의미하게 다르다는 것을 알 수 있다.

9. 지금까지 확인한 결과들을 모두 종합하여 고려한다면, boxplot에서 확인한 것과 같이 virginica, versicolor, setosa 순으로 평균이 높다고 할 수 있다.

[ML]

1. Creditcard 데이터를 df 데이터프레임에 불러오고, head(), describe(), info()를 통해 데이터 구조를 확인하였다.
2. Class가 0인 데이터는 df0으로, 1인 데이터는 df1로 분리하였다.
   df0의 데이터 중 10,000개를 무작위추출(random_state=42로 지정)하여 df0_sample에 저장하였다.
   df2에 df0_sample과 df1을 결합하여 저장하였다.
   df2의 Class별 개수와 비율을 확인한 결과,
   Class 0 은 10,000개, Class 1은 492개가 있음을 알 수 있었다.
   비율 또한 0.9531과 0.0468로 큰 차이가 있었다.
3. 'Amount' 열을 표준화하여 전처리하였다.
   전처리하고 Amount_Scaled 로 새로운 변수로 저장하였고, 기존 Amount 변수는 제거하였다.
   X는 df2에서 'Class' 변수를 제외한 모든 열, Y는 'Class' 변수만 포함하여 새로 만들었다.
4. 학습 데이터와 테스트 데이터로 분할하기 위해 test_size를 0.2로, train_size를 0.8

로 지정하였다. Stratify = Y로 지정하여 기존 Class의 비율을 유지하도록 하였다.
그 결과, Y_train에서 확인할 수 있는 데이터 개수는 약 8400개, Y_test는 약 2100개였다. 각 set 내 Class 0 과 1의 비율은 기존 set과 유사하게 약 0.95와 약 0.05로 구성되었다.

5. Y_train에서 Class 0이 약 8,000개 있는 것에 반해 Class 1은 약 400개 존재해 두 Class의 비율을 맞춰주기 위해 SMOTE, 오버샘플링하였다. 소수 클래스(1)에 대해 오버샘플링을 하여 사기 거래 건수를 추가로 만들어주었다.
   오버샘플링한 데이터를 new_X_train, new_Y_train으로 새롭게 저장하였고, Class의 개수를 확인한 결과 각 Class 모두 7,999개로 동일하다는 것을 알 수 있었다.
6. 데이터를 바탕으로 사기거래(Class=1)인지 아닌지(Class=0) 판단하여야 하기 때문에, Logistic Regression을 이용하였다.
   new_X_train과 new_Y_train을 이용하여 학습시켰다. 학습 데이터로 예측하였을 때 accuracy가 매우 높다는 것을 확인하였다.
   PR-AUR도 0.98로 매우 높았다.
7. 이제 만들어진 model을 이용해 test data를 가지고 성능을 평가하였다.
   (weighted avg 기준)
   recall의 값은 0.97, f1-score는 0.98였고, PR-AUC는 0.8823로 나왔다.
   성능을 개선하기 위해 GridSearch를 이용해 새롭게 모델을 만들었다.
   recall은 0.99, f1-score는 0.99, PR-AUC는 0.9048로 모든 목표를 통과하였다.