

기초 통계 / ML 과제

1. 기초 통계 분석 - Iris 데이터셋

1-1. 데이터 구조 확인

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   sepal_length 150 non-null   float64
1   sepal_width  150 non-null   float64
2   petal_length 150 non-null   float64
3   petal_width  150 non-null   float64
4   species      150 non-null   object
dtypes: float64(4), object(1)
```

1-2. 종별 기술통계량

종(Species)별 `petal_length`에 대해 평균, 표준편차, 사분위수 등을 계산해봤음.

	count	mean	std	min	25%	50%	75%	max
species								
setosa	50.0	1.462	0.173664	1.0	1.4	1.50	1.575	1.9

```
versicolor  50.0  4.260  0.469911  3.0  4.0  4.35  4.600  5.1
virginica    50.0  5.552  0.551895  4.5  5.1  5.55  5.875  6.9
```

Group sizes:

species

setosa 50

versicolor 50

virginica 50

Name: count, dtype: int64

→ virginica가 평균적으로 가장 길고, setosa가 가장 짧음.

1-3. Boxplot 시각화



→ petal_length의 평균은 virginica > versicolor > setosa으로 확인함.

1-4. 정규성 검정

각 품종별로 `petal_length`가 정규분포를 따르는지 Shapiro-Wilk 검정을 했음.

setosa: p-value = 0.0548
versicolor: p-value = 0.1585
virginica: p-value = 0.1098

→ 세 그룹 모두 $p > 0.05$ → 정규성을 만족

1-5. 등분산성 검정

세 그룹의 분산이 같은지 Levene 검정을 해봤음.

Levene Test p-value = 0.0000000313

→ $p < 0.05$ 여서 등분산성이 만족되지 않는다고 해석함.

1-6. ANOVA 분석

귀무가설 (H_0): 세 종(Species)의 Petal Length 평균은 모두 같음.
대립가설 (H_1): 적어도 하나의 종은 평균이 다름.

$F = 1180.1612$, $p = 0.0000$

→ $p < 0.05$ 로 귀무가설을 기각했고, 평균 차이가 있다고 판단함.

1-7. 사후검정 (Tukey HSD)

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
====
group1  group2  meandiff p-adj lower upper reject
-----
setosa versicolor  2.798  0.0 2.5942 3.0018  True
setosa virginica   4.09  0.0 3.8862 4.2938  True
versicolor virginica 1.292  0.0 1.0882 1.4958  True
-----
```

→ 세 그룹 모두 유의미한 차이가 있음을 확인함.

1-8. 결론

모든 그룹 쌍에서 평균 차이가 통계적으로 유의미함. setosa-versicolor, setosa-virginica, versicolor-virginica 모두 통계적으로 유의미한 차이가 있음.

→ 즉, 세 품종 간에 petal_length 평균이 모두 다르다.

2. 머신러닝 실습 - 신용카드 사기 탐지

2-1. 데이터 불러오기 및 탐색

신용카드 거래 데이터셋(`creditcard.csv`)을 불러와서 전체 크기와 클래스 비율을 확인해봤음.

```
(284807, 31)
```

```
Class
```

```
0    0.998273
```

```
1    0.001727
```

→ 전체 거래 중 사기 거래는 약 0.17%에 불과, 클래스 불균형이 심하여 바로 분석이 어려움.

2-2. 샘플링

전체 데이터를 그대로 쓰기에는 너무 크고, 정상 거래가 과도하게 많아서 샘플링을 진행했음.

사기 거래 492건은 전부 유지하고, 정상 거래는 10,000건만 랜덤 추출했음.

→ 클래스 비율이 약 **95.3% : 4.7%** 수준으로 완화되었음.

2-3. 데이터 전처리

`Amount` 변수는 범위가 크기 때문에 `StandardScaler` 로 표준화해줬고, 원래 `Amount` 는 제거함.

입력 데이터(X)와 타겟 변수(y)를 나눠준 뒤, 학습용과 테스트용으로 8:2 비율로 분할했음.

`stratify=y` 옵션을 줘서 클래스 비율은 그대로 유지함.

2-4. SMOTE 적용

학습 데이터셋에 대해 SMOTE(Synthetic Minority Over-sampling Technique)를 적용해서, 사기 거래 클래스(1)를 oversampling했음.

```
print("Before SMOTE:", y_train.value_counts())
print("After SMOTE:", pd.Series(y_train_resampled).value_counts())
```

Before SMOTE:

```
0    8000
1     398
```

After SMOTE:

```
0    8000
1    8000
```

→ 클래스가 균형을 이루도록 맞춰졌고, 모델이 소수 클래스를 무시하지 않도록 해줬음.

2-5. 모델 학습 및 성능 평가

`RandomForestClassifier` 를 사용해서 모델을 학습시켰고, 테스트셋에 대해 예측 후 평가 지표를 확인했음.

```
print(classification_report(y_test, y_pred, digits=4))
print(f"PR-AUC: {pr_auc:.4f}")
```

```
precision    recall  f1-score   support

0   0.9968   0.9956   0.9962     2000
1   0.8772   0.8891   0.8831      129

accuracy                0.9925     2129
macro avg   0.9370   0.9424   0.9396     2129
weighted avg   0.9926   0.9925   0.9926     2129

PR-AUC: 0.9581
```

→ Recall 0.8891, F1-score 0.8831, PR-AUC 0.9581로 꽤 좋은 성능을 보였음.

2-6. 결론

- 클래스 불균형을 그대로 두면 모델이 소수 클래스를 거의 예측하지 못하는데, SMOTE 덕분에 개선할 수 있었음.
- 모델 성능은 대부분 목표 ($\text{Recall} \geq 0.80$, $\text{F1} \geq 0.88$, $\text{PR-AUC} \geq 0.90$)를 만족했음.