1. Iris 데이터셋을 활용해 클래스별 변수 평균 차이를 검정

```
# 1. 데이터셋 불러오기
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

✓ 0.0s

	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

```
# 2. 기초통계량 산출
iris.groupby("species")["petal_length"].describe()
```

✓ 0.0s

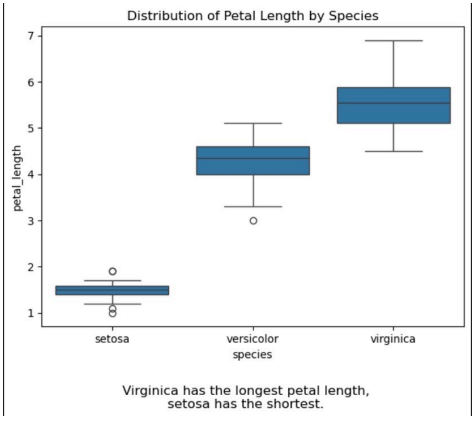
species	count	mean	std	min	25%	50%	75%	max
setosa	50.0	1.462	0.173664	1.0	1.4	1.50	1.575	1.9
versicolor	50.0	4.260	0.469911	3.0	4.0	4.35	4.600	5.1
virginica	50.0	5.552	0.551895	4.5	5.1	5.55	5.875	6.9

```
# 2. 그룹별 데이터 개수
iris["species"].value_counts()
```

✓ 0.0s

```
species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```

3. Species별 petal length 분포



4. 정규성검정

귀무가설(H_0): 각 species의 Petal Length는 정규분포를 따른다.
대립가설(H_1): 각 species의 Petal Length는 정규분포를 따르지 않는다.

setosa: p-value = 0.0548
귀무가설(H_0) 채택, p-value가 유의수준 0.05보다 크기 때문에 setosa의 Petal Length가 정규분포를 따른다고 볼 수 있다.

versicolor: p-value = 0.1585
귀무가설(H_0) 채택, p-value가 유의수준 0.05보다 크기 때문에 versicolor의 Petal Length가 정규분포를 따른다고 볼 수 있다.

virginica: p-value = 0.1098
귀무가설(H_0) 채택, p-value가 유의수준 0.05보다 크기 때문에 virginica의 Petal Length가 정규분포를 따른다고 볼 수 있다.

5. 등분산성 검정

귀무가설(H_0): 세 species의 Petal Length 분산은 모두 같다.
대립가설(H_1): 적어도 한 species의 Petal Length 분산이 다르다.

Levene 등분산성 검정 p-value = 3.1287566394085344e-08
귀무가설(H_0) 기각, p-value가 유의수준 0.05보다 작기 때문에 등분산성을 만족한다고 보기 어렵다.

6. 가설 수립

귀무가설(H_0): 3개 species의 평균 petal_length는 모두 같다.
대립가설(H_1): 적어도 한 species의 평균 petal_length는 다르다.

7. ANOVA

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

ANOVA p-value: 2.8568e-91
귀무가설(H_0) 기각: 3개 종의 평균이 통계적으로 유의미하게 다르다.

8. 사후검정

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

9. 결과 요약

Shapiro-Wilk 검정 결과, 세 집단(setosa, versicolor, virginica)의 Petal Length는 모두 정규성을 만족하였다($p > 0.05$). 반면, Levene 검정을 통해 등분산성이 통계적으로 유의하게 성립하지 않음($p < 0.05$).

이러한 조건 하에 수행된 ANOVA 결과, 세 집단 간 평균 Petal Length에 유의한 차이가 있다.
사후검정으로 수행한 Tukey의 HSD 결과, 각 집단 간 쌍 비교 모두에서 통계적으로 유의한 차이가 존재하였다으며, 세 집단의 평균 Petal Length는 virginica > versicolor > setosa 순으로, virginica가 가장 길고 setosa가 가장 짧았다.
이와 같은 결과는 시각화된 Boxplot 및 Tukey HSD 결과와 일관된 양상을 보이며, 세 품종 간의 명확한 형태적 차이를 통계적으로도 뒷받침한다.

2. 실제 신용카드 사기 데이터셋을 활용해 클래스 불균형 상황에서 분류 모델을 학습

1. 데이터 로드 및 기본 탐색

```
   Time        V1        V2        V3        V4        V5        V6        V7  \
0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599
1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461
3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609
4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941

         V8        V9  ...       V21       V22       V23       V24       V25  \
0  0.098698  0.363787  ... -0.018307  0.277838 -0.110474  0.066928  0.128539
1  0.085102 -0.255425  ... -0.225775 -0.638672  0.101288 -0.339846  0.167170
2  0.247676 -1.514654  ...  0.247998  0.771679  0.909412 -0.689281 -0.327642
3  0.377436 -1.387024  ... -0.108300  0.005274 -0.190321 -1.175575  0.647376
4 -0.270533  0.817739  ... -0.009431  0.798278 -0.137458  0.141267 -0.206010

        V26       V27       V28  Amount  Class
0 -0.189115  0.133558 -0.021053  149.62      0
1  0.125895 -0.008983  0.014724    2.69      0
2 -0.139097 -0.055353 -0.059752  378.66      0
3 -0.221929  0.062723  0.061458  123.50      0
4  0.502292  0.219422  0.215153   69.99      0

[5 rows x 31 columns]
Class
0    284315
1       492
Name: count, dtype: int64
```

2. 샘플링

```
Class
0    10000
1      492
Name: count, dtype: int64
Class
0    95.310713
1     4.689287
Name: proportion, dtype: float64
```

4. 학습 데이터와 테스트 데이터 분할

```
Train set class distribution:
Class
0    7999
1     394
Name: count, dtype: int64
Class
0    95.305612
1     4.694388
Name: proportion, dtype: float64

Test set class distribution:
Class
0    2001
1      98
Name: count, dtype: int64
Class
0    95.33111
1     4.66889
Name: proportion, dtype: float64
```

5. SMOTE 적용

```
Before SMOTE:
Class
0    7999
1     394
Name: count, dtype: int64

After SMOTE:
Class
1    7999
0    7999
Name: count, dtype: int64
```

SMOTE 적용의 근거

SMOTE를 적용하는 이유는, 현재 학습데이터에서 사기거래의 수가 현저히 적기 때문입니다. 이처럼 클래스 간 불균형이 심한 데이터를 그대로 학습에 적용하면, 모델은 대부분의 경우를 정상 거래오 예측하게 되기 때문에, 실제로 중요한 사기 거래를 제대로 탐지하지 못하는 문제가 발생합니다. SMOTE를 활용하여 기존의 사기 거래 데이터와 유사한 새로운 가상의 데이터를 생성함으로써 이러한 문제점을 해결할 수 있습니다.

6. 모델 학습

```
=== Classification Report ===
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      2001
           1       0.81      0.93      0.86        98

    accuracy                           0.99      2099
   macro avg       0.90      0.96      0.93      2099
weighted avg       0.99      0.99      0.99      2099


=== PR-AUC (Precision-Recall AUC) ===
0.9534699504261935
```

7. 최종 성능 평가

데이터 셋이 범주형 자료이며 SMOTE를 활용하여 범주간의 데이터 개수의 차이를 맞추어 주었기에 가장 기본이 되는 로지스틱 회귀를 모델로 선정하였습니다.
테스트셋 기준으로 Recall(0.93), PR-AUC(0.95)는 달성하였지만 F1-score(0.86)으로 달성하지 못하였습니다. 하지만 전체적으로 모델은 사기 거래를 잘 탐지하고 있으며 우수한 성능을 보였습니다. 목표한 F1-score를 높이기 위해서는 예측 확률을 threshold 조정해주는 방법이 있습니다.