유방암 여부 로지스틱 회귀 예제

라이브러리 및 패키지 Import

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

# 경고 무시 (FutureWarning)
import warnings
warnings.filterwarnings('ignore', category=FutureWarning)
```

데이터셋 불러오기

In [2]:
```
# 유방암 데이터셋 로드
cancer = load_breast_cancer()
df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
df['target'] = cancer.target
```

In [3]:
```
# 데이터프레임 확인
df.head()
```

Out[3]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...

5 rows × 31 columns

데이터 전처리

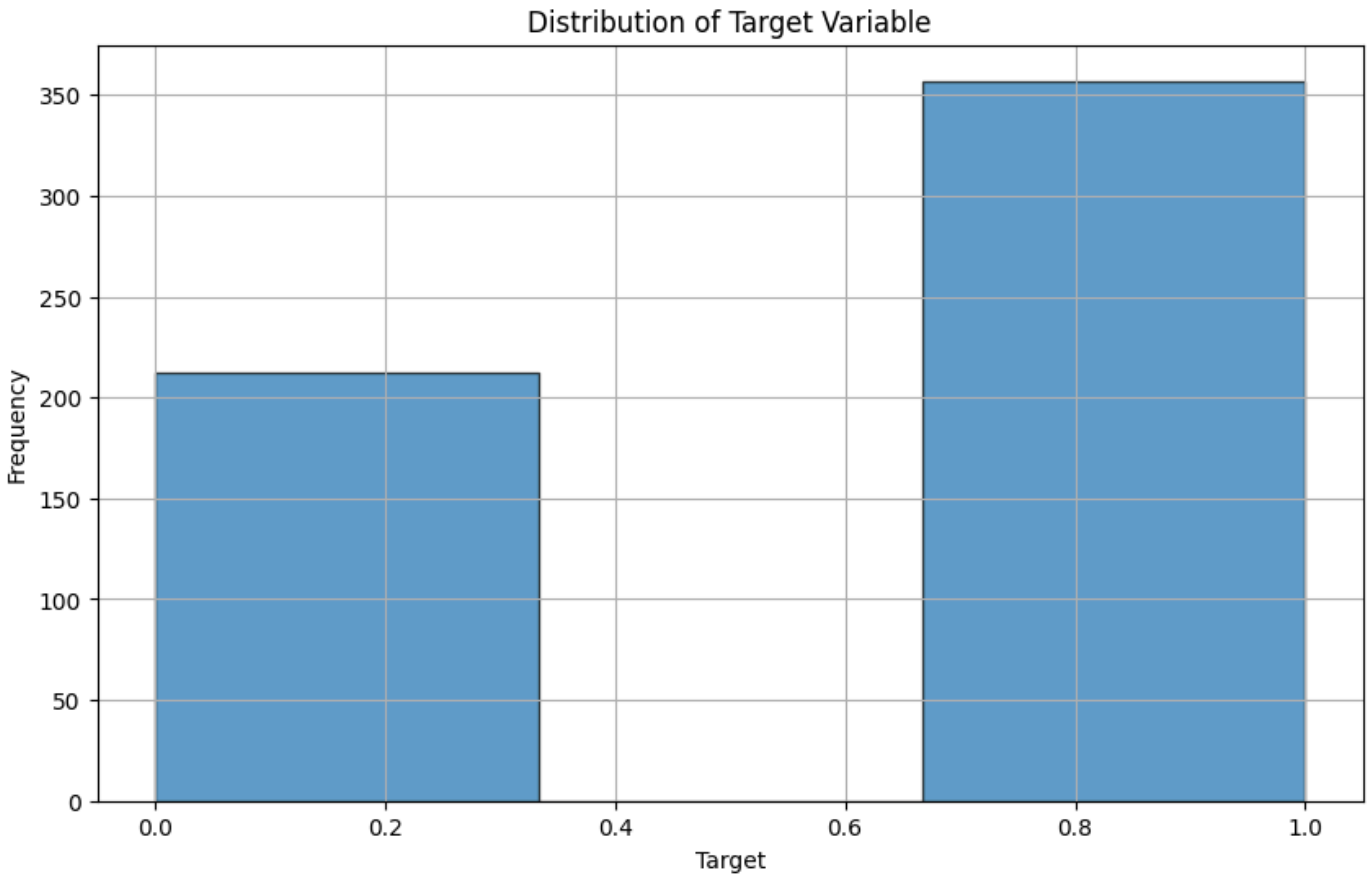
In [4]:
```
# 데이터 전처리
X = df.drop(columns=['target'])
y = df['target']

# 훈련 세트와 테스트 세트로 분리
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

In [5]:
```
# 피처 스케일링
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

데이터 분포 확인

In [6]:
```
# 목표 변수 데이터 분포 확인
plt.figure(figsize=(10, 6))
plt.hist(df['target'], bins=3, edgecolor='black', alpha=0.7)
plt.title('Distribution of Target Variable')
plt.xlabel('Target')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



로지스틱 회귀 모델 학습

In [7]:
```
# 로지스틱 회귀 모델 학습
model = LogisticRegression()
model.fit(X_train, y_train)

# 테스트 세트로 예측 수행
y_pred = model.predict(X_test)
```

In [8]:
```
# 모델 평가
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(conf_matrix)
print('Classification Report:')
print(class_report)
```

```
Accuracy: 0.9736842105263158
Confusion Matrix:
[[41  2]
 [ 1 70]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.95      0.96        43
           1       0.97      0.99      0.98        71

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114
```

In [12]:
```
# 각 Feature의 계수 확인(영향도 분석)
print("회귀계수(coefficients):", np.round(model.coef_, 2))
```

```
회귀계수(coefficients): [[-0.43 -0.39 -0.39 -0.46 -0.07  0.54 -0.8  -1.12  0.24  0.08 -1.
27  0.19
  -0.61 -0.91 -0.31  0.69  0.18 -0.32  0.5   0.61 -0.88 -1.34 -0.59 -0.85
  -0.55  0.01 -0.95 -0.77 -1.21 -0.15]]
```

결과 시각화

In [11]:
```
# 혼동 행렬 시각화
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=cancer.target_na
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```