로지스틱 회귀

로지스틱 회귀

- 로지스틱 회귀 (Logistic Regression)
  - 선형 회귀 모형을 '분류'에 적용한 기법
  - 데이터가 특정 레이블(클래스)에 소속될 확률을 추정한다.

    예1) 이 이메일이 스팸일 확률은 얼마인가?

    예2) 이번 시험에서 합격할 확률은 얼마인가?
  - 다른 선형 회귀 모형과는 다르게, 종속변수가 수치형(numerical)이 아니라 범주형(categorical)이다.

    예1) 스팸메일, 정상메일

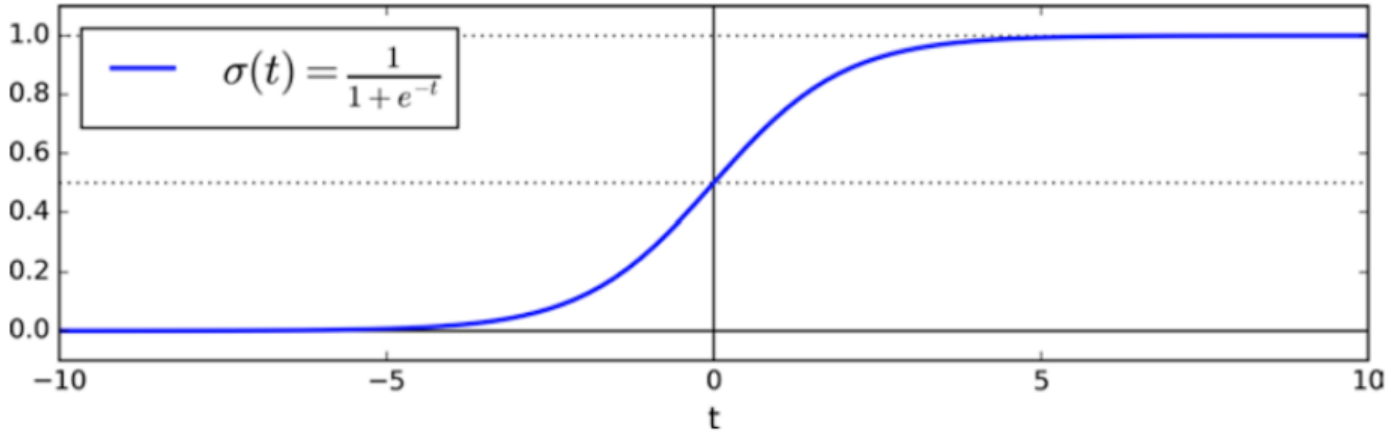
    예2) 합격, 불합격
  - 특정 클래스에 대해서 추정된 확률이 50% 이상이면 해당 데이터를 그 클래스에 속하는 것으로 분류한다.

로지스틱 회귀

- 로지스틱 회귀
  - 기본적인 로지스틱 회귀는 이항형(binomial)으로서, 종속변수의 값의 종류는 0과 1의 두 종류 뿐이다.
    - 즉, 이 경우의 종속변수는 곧 클래스 그 자체이다.
    - 값이 0이면 음성, 1이면 양성이라고 표현할 수 있다.
  - 이러한 이진 데이터에 대해서 올바른 결과를 나타내는 선형 회귀를 수행하려면 다음과 같은 성질이 필요하다.
    - 연속적인 단조 증가(monotone increasing) 함수일 것
    - 함수의 결과가 [0, 1] 사이의 값일 것
  - 이와 같은 성질을 만족하는 함수를 시그모이드(sigmoid) 함수라고 한다.

로지스틱 회귀

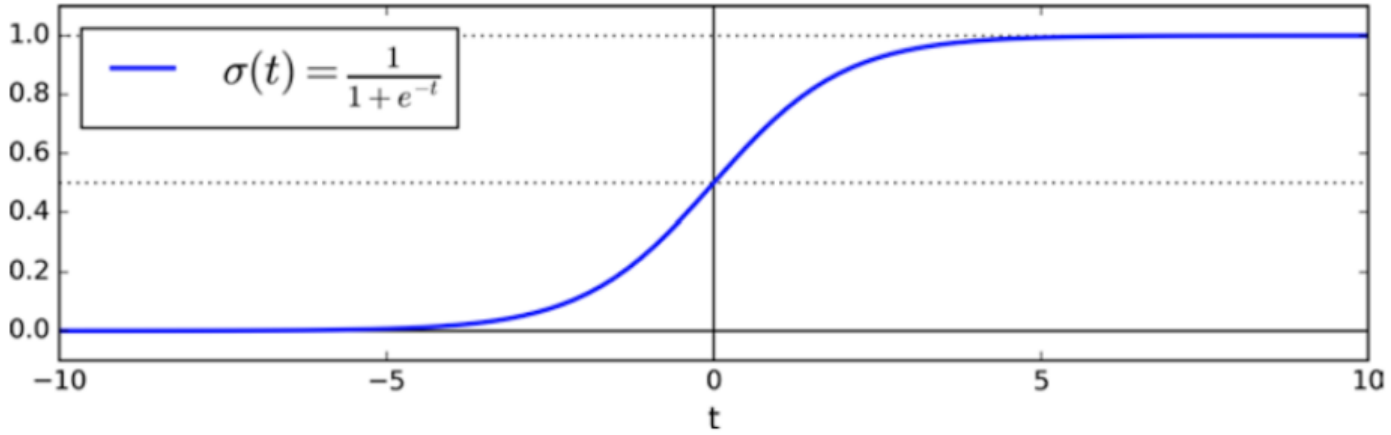
- 시그모이드 함수와 확률 추정
  - 시그모이드 함수 (로지스틱 함수)



  - 예측값 t가 시그모이드 함수에 전달되면 결과 값 σ(t)가 계산되며, 이 값은 추정된 확률 p이다. ($0 \leq p \leq 1$)

로지스틱 회귀

- 시그모이드 함수와 확률 추정
  - 시그모이드 함수 (로지스틱 함수)



  - 추정된 확률 p의 값이 0.5 이상이면 결과를 1으로 분류하고 p의 값이 0.5 미만이면 0으로 분류한다.
  - 즉, $t \geq 0$ 이면 결과는 양성, $t < 0$ 이면 결과는 음성이다.

로지스틱 회귀

- 데이터 선정 및 분포 정보 확인
  - 붓꽃 데이터를 불러와서 정보를 확인한다.

```
1  import sklearn.datasets as d
2
3  iris = d.load_iris()
4  print(iris.DESCR)
```

```
**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica
```

여기에서는 결과의 일부만 보였다.

로지스틱 회귀

- 데이터 선정 및 분포 정보 확인
  - 붓꽃 데이터를 불러와서 정보를 확인한다.

```
1  print("특성 :", iris.feature_names)
2  print("클래스 :", iris.target_names)
```

```
특성 : ['sepal length (cm)', 'sepal width (cm)',
        'petal length (cm)', 'petal width (cm)']
클래스 : ['setosa' 'versicolor' 'virginica']
```

```
1  print("특성 자료형 :", type(iris.data))
2  print("클래스 자료형 :", type(iris.target))
```

```
특성 자료형 : <class 'numpy.ndarray'>
클래스 자료형 : <class 'numpy.ndarray'>
```

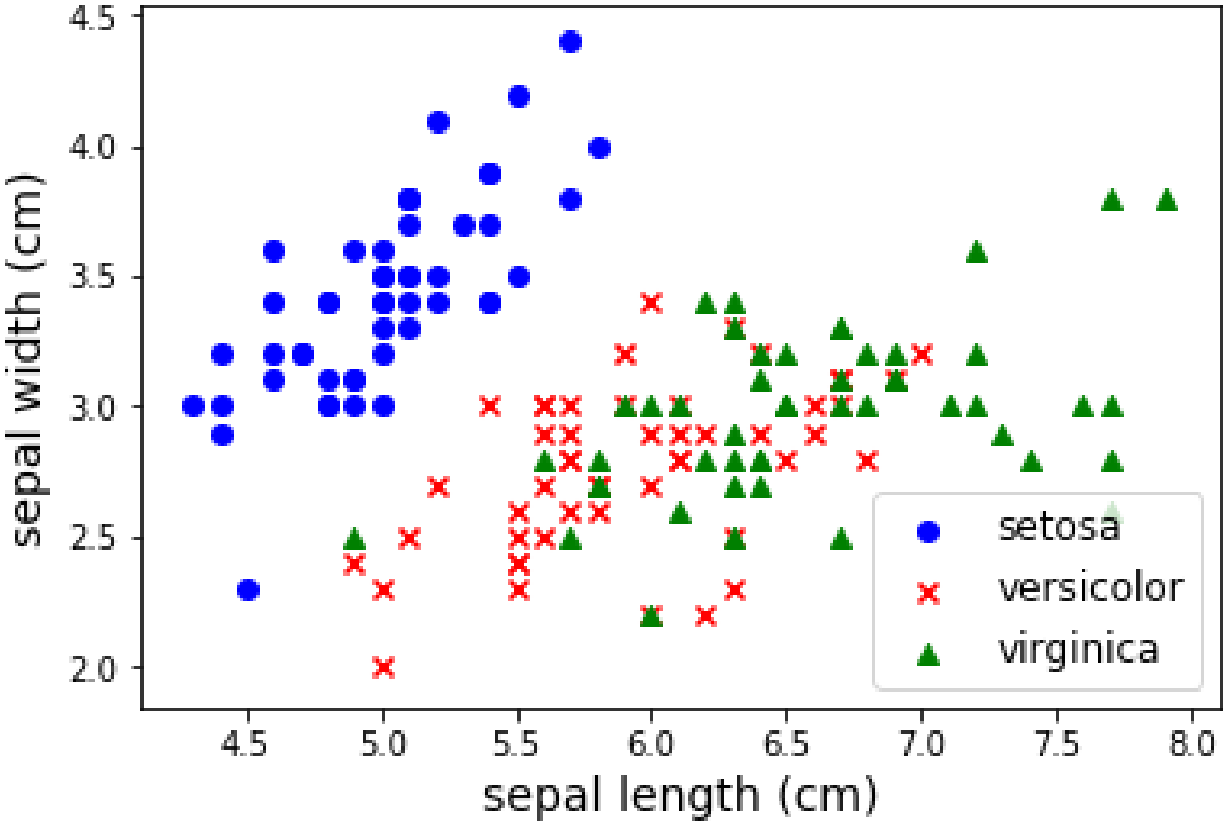
로지스틱 회귀

- 데이터 선정 및 분포 정보 확인
  - 꽃받침의 길이와 너비 컬럼을 선택하여 산점도를 그린다.

```
import matplotlib.pyplot as plt

c_set = ["blue", "red", "green"]
m_set = ["o", "x", "^"]
l_set = iris.target_names

X = iris.data
Y = iris.target

for t in set(Y):
    X_pts = [X[i, 0] for i in range(len(Y)) if Y[i] == t]
    Y_pts = [X[i, 1] for i in range(len(Y)) if Y[i] == t]
    plt.scatter(X_pts, Y_pts, color=c_set[t], marker=m_set[t], label=l_set[t])

plt.xlabel(iris.feature_names[0], fontsize=14)
plt.ylabel(iris.feature_names[1], fontsize=14)
plt.legend(loc='best', fontsize=12)
```

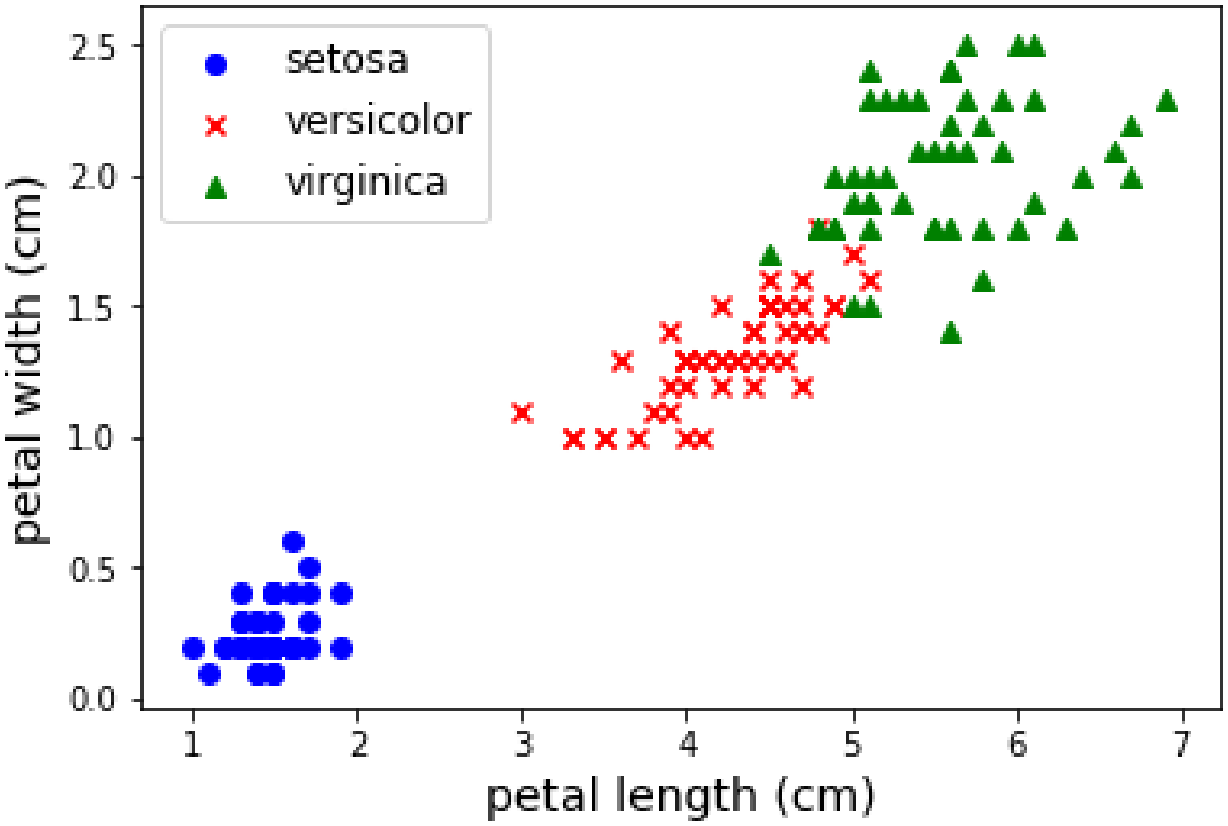
로지스틱 회귀

- 데이터 선정 및 분포 정보 확인
  - 산점도를 바탕으로, 각 클래스 분포를 확인한다.



로지스틱 회귀

- 데이터 선정 및 분포 정보 확인
  - 동일한 방식으로 꽃잎의 길이와 너비 컬럼에 대해 산점도를 그려서 각 클래스 분포를 확인한다.



로지스틱 회귀

- 사이킷런으로 로지스틱 회귀 수행
  - 데이터를 학습용과 검증용으로 분리한다.

```
1  import sklearn.model_selection as ms
2
3  X_ptls = X[:, 2:4]
4  Y_vgnc = (Y == 2).astype(np.int)
5
6  X_train, X_test, y_train, y_test = ₩
7  ms.train_test_split(X_ptls, Y_vgnc, test_size=0.3, random_state=42)
```

```
1  X_train[0:3,]
```

```
array([[3.7, 1. ],
       [5.1, 1.5],
       [5.5, 1.8]])
```

```
1  y_train[0:3,]
```

```
array([0, 1, 1])
```

로지스틱 회귀

- 사이킷런으로 로지스틱 회귀 수행
  - linear_model 모듈에 있는 LogisticRegression을 이용하여 로지스틱 회귀를 수행한다.
    - 매개변수 solver는 회귀를 수행할 알고리즘의 이름이다. 기본값은 'liblinear'이다.

```
1  import sklearn.linear_model as lm
2
3  logr = lm.LogisticRegression(solver="liblinear")
4  reg = logr.fit(X_train, y_train)
```

※ 매개변수 solver의 기본값이 추후 사이킷런 0.22 버전에서 'lbfgs'로 변경될 예정이다.

로지스틱 회귀

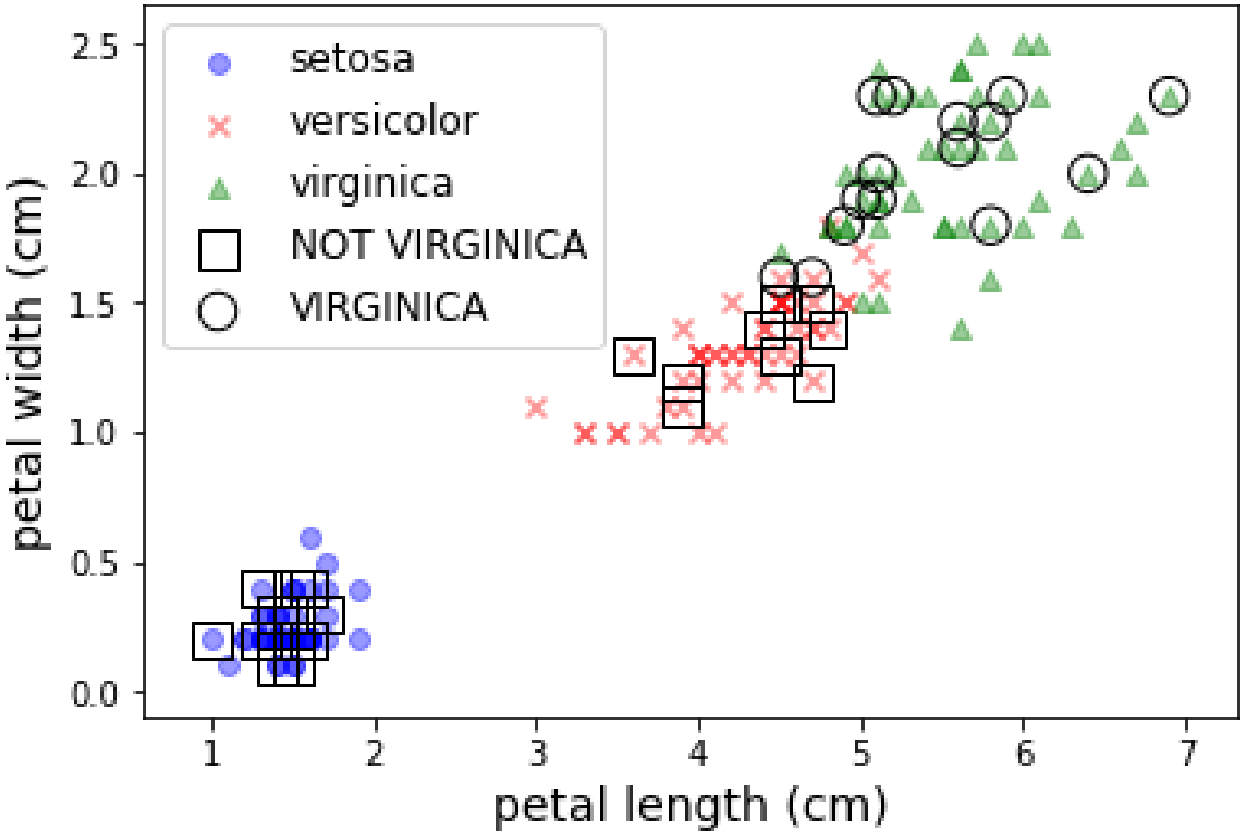
- 사이킷런으로 로지스틱 회귀 수행
  - 검증 데이터로 예측을 수행한 뒤, **metrics** 모듈에 있는 **accuracy_score** 함수를 이용하여 정확도를 계산한다.
    - 이 때 첫 번째 매개변수는 검증 데이터의 클래스 실제값이고, 두 번째 매개변수는 클래스 예측값이다.

```
1  import sklearn.metrics as mt
2
3  y_pred = reg.predict(X_test)
4
5  accuracy = mt.accuracy_score(y_test, y_pred)
6  print("예측 정확도:", round(accuracy, 3))
```

```
예측 정확도: 0.956
```

로지스틱 회귀

- 사이킷런으로 로지스틱 회귀 수행
  - 검증 데이터의 예측 결과에 대한 분포를 확인한다.



로지스틱 회귀

- 로지스틱 회귀의 특징
  - 분석 기법의 이름 자체에 '회귀'라는 단어가 들어가 있지만 다른 회귀 모형의 목적과 다르게 '분류'를 수행한다.
  - 즉, 로지스틱 회귀 모형은 분류기(classifier)이다.
  - 로지스틱 회귀의 기본 형태는 이진 분류이며, 예측 성능도 일반적으로 우수한 편이다.
  - 실행 알고리즘을 다른 것으로 변경하여 다중 클래스들의 분류를 수행할 수 있다.

로지스틱 회귀

- 다중 클래스 분류를 위한 로지스틱 회귀
  - 다중 클래스들에 대한 분류를 수행하는 회귀 모형을 다항(multinomial) 로지스틱 회귀 또는 소프트맥스(softmax) 회귀라고 한다.
  - 소프트맥스 회귀를 수행하려면 LogisticRegression의 매개변수 multi_class를 'multinomial'로, 매개변수 solver를 'lbfgs'로 지정해 준다.

```
1  import sklearn.linear_model as lm
2
3  smr = lm.LogisticRegression(multi_class="multinomial", ₩
4                              solver="lbfgs")
5  reg = smr.fit(X_train, y_train)
```