회귀분석(Linear Regression Analysis):

y(종속변수, 출력변수)가 x(독립변수, 입력변수)에 의하여 설명된다고 보고 그 함수 관계를 조사하는 통계적인 기법.

선형회귀분석(Linear Regression Analysis):회귀분석에서 y(종속변수, 출력변수)와 x(독립변수, 입력변수)의 함수관계가 선형(linear).

단순선형회귀분석(Simple Linear Regression Analysis): x(독립변수, 입력변수)의 개수가 1인 선형회귀분석.

중선형회귀분석(Multiple Linear Regression Analysis): x(독립변수, 입력변수)의 개수가 2개 이상인 선형회귀분석

중선형회귀분석(Multiple Linear Regression Analysis)

체중(x1), 나이(x2)와 혈압(y)과의 관계조사

공정온도(x1), 공정압력(x2)과 제품의 강도(y)와의 관계조사

x : 독립변수(independent variable) or 설명변수(explanatory variable)가 2개 이상

y : 종속변수(dependent variable) or 반응변수(response variable)

중선형회귀모형(Multiple Linear Regression Model)

x1, x2 와 y의 관찰값이 관계가 선형 such that

$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + e_i \quad i = 1,2,,\cdots,n \quad e_i \sim iid\ N(0, \sigma^2)$

단순선형회귀분석(Simple Linear Regression Analysis)

eg) 약의 복용량(x)과 약효지속시간(y)의 관계조사

공정온도(x)와 제품의 강도(y)관계조사

TOEFL점수(x)와 학과 영어과목성적(y)과의 관계

x : 독립변수(independent variable), 설명변수(explanatory variable)

입력변수(input variable)

y : 종속변수(dependent variable), 반응변수(response variable)

출력변수(output variable), target 변수

1. 단순선형회귀모형(Simple Linear Regression Model)

2변수의 x와 y의 관찰값이 관계가 다음과 같은 경우:

$y_i = \beta_0 + \beta_1 x_i + e_i \quad i = 1,2,,\cdots,n \quad e_i \sim iid\ N(0, \sigma^2)$

β_0 : y와 x의 직선의 절편(intercept)을 나타내는 모수(parameter)

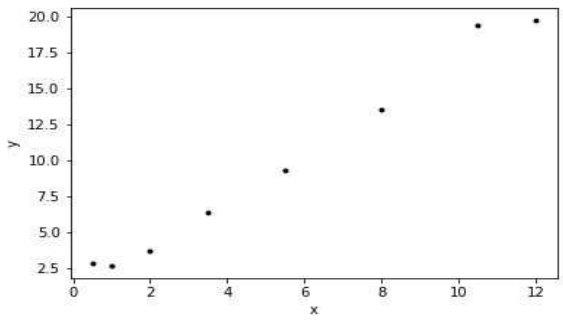
β_1 : y와 x의 직선의 기울기(slope)를 나타내는 모수(parameter)

~ x가 1증가하면 y는 β_1만큼 증가

y_i : i번째 y의 값, x_i : i번째 실험의 x의 값(입력변수)

eg1) 약의 복용량(x)과 약효지속시간(y)의 관계조사

i	x_i (mg)	y_i (hr)
1	0.5	2.86
2	1.0	2.66
3	2.0	3.69
4	3.5	6.40
5	5.5	9.27
6	8.0	13.53
7	10.5	19.38
8	12.0	19.71



2. 최소제곱법(Least Square Method)

주어진 데이터 $(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$를 이용하여

β_0,β_1 의 추정값인 $\hat{\beta}_0,\hat{\beta}_1$ 를 구하기

$\hat{\beta}_0,\hat{\beta}_1$은 $Q=\sum_{i=1}^{n}(y_i-b_0-b_1x_i)^2$ 를 최소화시키는 b_0,b_1의 값

→ $\hat{\beta}_1$: $\frac{\partial Q}{\partial b_1}=0$ 를 만족시키는 b_1의 값 , $\hat{\beta}_0$: $\frac{\partial Q}{\partial b_0}=0$ 를 만족시키는 b_0의 값

$\equiv X=(\mathbf{1},\boldsymbol{x}),\ \mathbf{1}\in R^{n\times 1},\ \boldsymbol{x}\in R^{n\times d}, d=1,\ \boldsymbol{y}\in R^{n\times 1}$

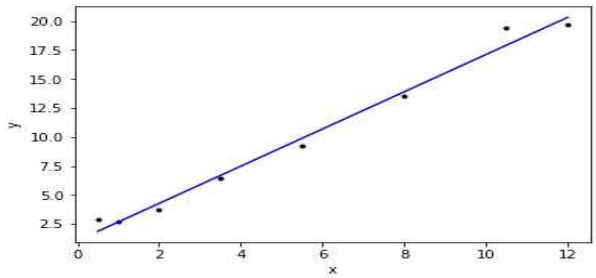
$$\hat{\boldsymbol{\beta}}=\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix}=(X'X)^{-1}X'\boldsymbol{y} \qquad \text{(A)}$$

* 적합된 회귀직선(Fitted regression line) : $\hat{y}=X\hat{\boldsymbol{\beta}}=\hat{\beta}_0+\hat{\beta}_1x$ (B)

eg) $\hat{\boldsymbol{\beta}}=\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix}=\begin{pmatrix}1.0605\\ 1.6046\end{pmatrix}$

적합된 회귀직선(Fitted regression line) : $\hat{y}=\hat{\beta}_0+\hat{\beta}_1x=-1.0605+1.6046x$ =

~약복용량이 1mg 증가하면 약효지속시간은 1.6046시간 증가.



eg2) 시험자료가 주어지면, $\boldsymbol{x}_t=(1.5,\ 3,\ 6,\ 9,\ 13)'$ ~ $\boldsymbol{X}_t=(\mathbf{1},\boldsymbol{x}_t),\ \mathbf{1}\in R^{5\times 1}$

약효지속시간의 예측값은 $\hat{y}(\boldsymbol{x}_t)=X_t\hat{\boldsymbol{\beta}}=\hat{\beta}_0+\hat{\beta}_1x_t=-1.0605+1.6046x_t$ (C)

3. 결정계수

SSTO(Total Sum of Squares, 전체제곱합) = $\sum_{i=1}^{n}(y_i-\bar{y})^2$

SSE(Error Sum of Squares, 오차제곱합) = $\sum_{i=1}^{n}\hat{e}_i=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

MSE(Mean Squared errors) = $\frac{SSE}{n-2}$: σ^2 의 추정값 = $\hat{\sigma}^2$

SSR(Refression Sum of Squares, 회귀제곱합) = $\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2$ = SSTO-SSE

결정계수(R^2 coefficient of determination): x변수를 사용함으로써 줄어드는 전체의 변동의 비율 - x변수가 설명할수 있는 y변수의 변동의 비율 - 주어진 모델이 전체변동(y)를 얼마나 잘 설명하는지 나타내는 값 [0,1]

$$R^2=\frac{SSR}{SSTO}$$

x와 y의 상관계수(r, Correlation coefficient) = $\pm\sqrt{R^2}$ (부호= $\hat{\beta}_1$의 부호) [- or +]

<Python>

```
import numpy as np
import matplotlib.pyplot as plt
n=8; nt=5
x=np.array([ 0.5,  1. ,  2. ,  3.5,  5.5,  8. , 10.5, 12. ]).reshape(n,1)
y=np.array([ 2.8654883 ,  2.66422646,  3.68812138,  6.39507074,  9.26655548, 13.52528373,
19.38166085, 19.6975463 ]).reshape(n,1)
xt=np.array([ 1.5 ,  3,  6,  9, 13]).reshape(nt,1)

X=np.hstack((np.ones((n,1)),x))
beta=np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)  #(A)
print('beta=',beta)

yh=X.dot(beta)    #(B)
e=y-yh # 잔차(residual)
SSTO=np.sum((y-np.mean(y))**2)
SSE=np.sum(e**2)
SSR=SSTO-SSE
R2=SSR/SSTO
corr=np.sign(beta[1])*np.sqrt(R2)
corr1=np.corrcoef(x.T,y.T) # corr=corr1[0,1]
print('R square=',R2,'cor(x,y)=', corr)
print('cor(x,y) matrix=',corr1)

plt.figure(1)
plt.plot(x,y,'k.');plt.plot(x,yh,'b-')
plt.xlabel('x'); plt.ylabel('y')

Xt=np.hstack((np.ones((nt,1)),xt))
yth=Xt.dot(beta)        #(C)
print('yth=',yth)
```

<싸이킷런 이용>

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

n=8; nt=5
x=np.array([ 0.5,  1. ,  2. ,  3.5,  5.5,  8. , 10.5, 12. ]).reshape(n,1)
y=np.array([ 2.8654883 ,  2.66422646,  3.68812138,  6.39507074,  9.26655548, 13.52528373,
19.38166085, 19.6975463 ]).reshape(n,1)
xt=np.array([ 1.5 ,  3,  6,  9, 13]).reshape(nt,1)

model = LinearRegression().fit(x, y)
beta0=model.intercept_ # β̂0
beta1=model.coef_      # β̂1
```

```
R2= model.score(x,y)   #R^2
yh=model.predict(x)    #ŷ = ŷ(x)
yth=model.predict(xt)  #ŷ(x_t)

plt.figure(1)
plt.plot(x,y,'k.')
plt.plot(x,yh,'b-')
plt.xlabel('x'); plt.ylabel('y')
```

Decision Tree (의사결정나무)

자료의 이해와 해석이 쉬운게 장점.

eg) 붓꽃 자료(자료수n=n1+n2+n3=50+50+50)

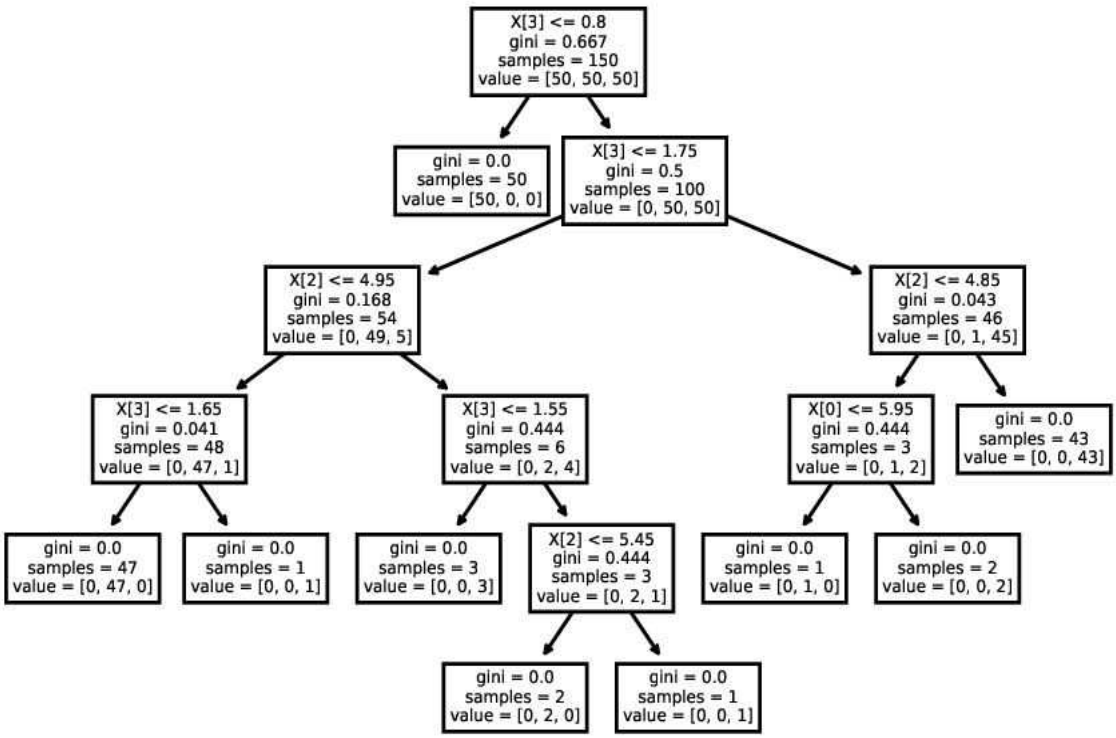
클래스1, 클래스2, 클래스3 = Setosa, Versicolor, Virginica~ 3품종

변수=SL, SW, PL, PW ~ 꽃받침길이, 꽃받침너비, 꽃잎길이, 꽃잎너비

```
import numpy as np
import matplotlib.pyplot as plt
# 붓꽃자료 불러오기
from sklearn.datasets import load_iris
iris = load_iris()
print(iris.feature_names) # 변수명: (X[0],X[1],X[2],X[3])=(SL,SW,PL,PW)
X, Y = load_iris(return_X_y=True)

from sklearn import tree # DecisionTree 불러오기
clf = tree.DecisionTreeClassifier(max_depth=5).fit(X,Y)
plt.figure(0)
DTree5=tree.plot_tree(clf.fit(X,Y))#iris.data, iris.target))
#그림(DTree5)을 eps파일로 저장 (jpg등도 가능, eps가 화질 좋음)
plt.savefig('DTree5.eps')

#깊이(depth)=3 그림
clf3 = tree.DecisionTreeClassifier(max_depth=3).fit(X,Y)
yh3=clf3.predict(X)
plt.figure(1)
DTree3=tree.plot_tree(clf3.fit(X,Y))
#그림(DTree3)을 eps파일로 저장
plt.savefig('DTree3.eps')
```



Tree 구조(Dtree1): 깊이=5.

지니불순도(Gini impurity): 자료에 클래스가 균일하게 섞여 있으면 커짐.

$gini = 1 - \sum_{k=1}^{m} p(k|t)^2$, $p(k|t)$: t번째 노드(node)에서 클래스k에 속하는 샘플의 비율

원자료(root node): 150개 샘플에서 각클래스 샘플수=50:50:50

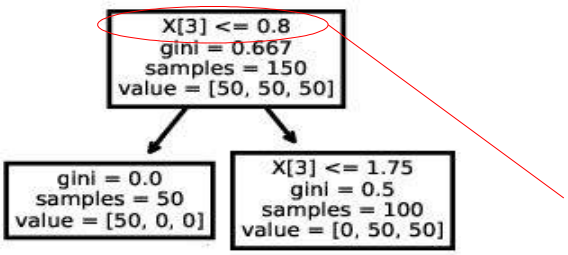
$gini = 1 - \sum_{k=1}^{3} p(k|0)^2 = 1 - \sum_{k=1}^{3} (50/150)^2 = 1 - 1/3 = 0.667$

여러 가지 분류기준 중 지니불순도를 최소화하는 분류기준 구하기

→ 분류기준: $X_3 = 0.8$, 이때 $gini$=0+0.5=0.5 ~ 분류기준이하가 yes는 왼쪽밑으로, 아니면 오른쪽밑으로.

$X_3 \leq 0.8$ 되는(왼쪽밑) 샘플수=50, 클래스 샘플수=50:0:0, 이때 지니불순도(gini)=0 -- stop

$X_3 > 0.8$ 되는(오른쪽밑) 샘플수=100, 클래스 샘플수=0:50:50, 이때 $gini$=0.5



다음 자료분할시 분류기준이 X[3]=0.8,

0.8이하면 왼쪽밑, 0.8보다 크면 오른쪽밑으로

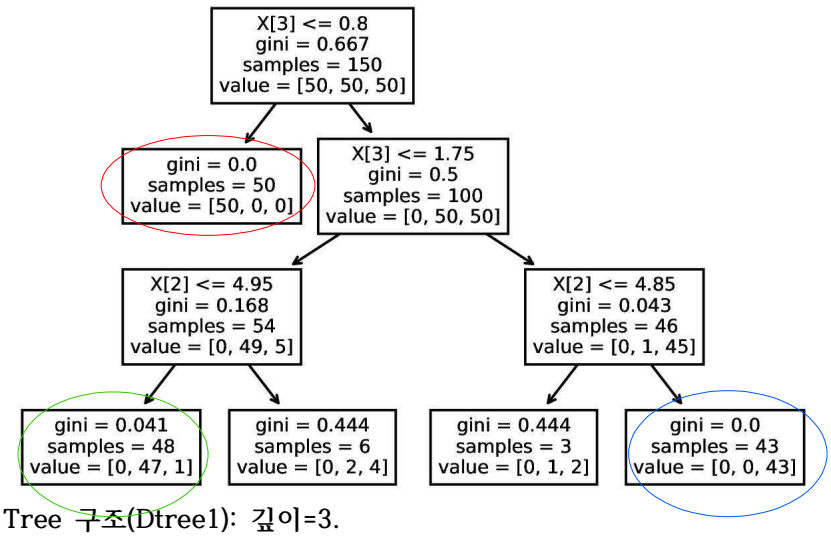
$X_3 > 0.8$ 되는 샘플에서 여러 가지 분류기준중 지니불순도를 최소화하는 분류기준 구하기

→ 분류기준: $X_3 = 1.75$, 이때 $gini$=0.168+0.043=0.211

$X_3 > 0.8$ & $X_3 \leq 1.75$ 되는 샘플수=54, 클래스 샘플수=0:49:5이때 $gini$=0.168

$X_3 > 0.8$ & $X_3 > 1.75$ 되는 샘플수=46, 클래스 샘플수=0:1:45이때 $gini$=0.043

시험자료가 주어진 경우(좀더 일반화있게 분류, 훈련자료가 너무 정확히 분류되면 시험자료 분류가 정확해지지 않음~ overfitting)



Tree 구조(Dtree1): 깊이=3.

1. 만약 X[3]≤0.8 ==> 클래스1(Setosa) (red동그라미)
2. 만약 X[3]>0.8 & X[3]≤1.75 & X[2]≤4.95 ==> 클래스2(Versicolor)(green동그라미)

== 만약 X[2]≤4.95 & 0.8<X[3]≤1.75 ==> 클래스2(Versicolor)

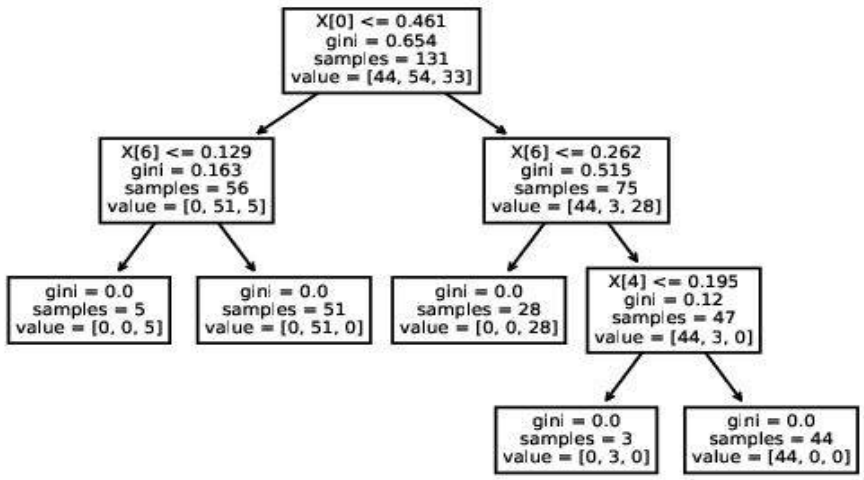
3. 만약 X[3]>0.8 & X[3]>1.75 & X[2]>4.85 ==> 클래스3(Virginica)(blue동그라미)

== 만약 X[3]>1.75 & X[2]>4.85 ==> 클래스3(Virginica)

RF(random forest):
여러 개의 의사결정나무(decision trees) 모은 것.
자료가 주어지면 Bootstrap sample(복원추출 표본)을 여러 개 추출해서 각 자료에 decision tree를 적용하여 각 결과의 평균을 결과로. 분류문제에 많이 사용됨.

```
#와인자료 : Italy의 같은 지역에서 재배된 3가지 품종(1,2,3)의 포도로 만든 와인의 분류
import numpy as np
import pandas as pd
from sklearn import tree
yx=pd.read_csv('d:/Python2021_Lecture/wine_tr.csv')
yx=yx.values
y=yx[:,0]; x=yx[:,1:] # 12개 변수(X[0]~X[11])
M = tree.DecisionTreeClassifier().fit(x,y)
print(M.feature_importances_) # 12개 변수 중요도
 결과: [0.44283579  0.       0.       0.       0.06558198  0.
        0.38525106  0.       0.       0.       0.          0.10633117]
~ 중요변수 = X[0], X[6], X[11], X[4]
yh=M.predict(x)

plt.figure(0)
DTree_wine=tree.plot_tree(M.fit(x,y))
#그림(DTree_wine)을 eps파일로 저장 (jpg등도 가능)
plt.savefig('DTree_wine.eps')
```



1. 만약 X[0]<=0.461 & X[6]<=0.129 ==> 품종3
2. 만약 X[0]<=0.461 & X[6]>0.129 ==> 품종2
3. 만약 X[0] >0.461 & X[6]<=0.262 ==> 품종3
4. 만약 X[0] >0.461 & X[6] >0.262 & X[4]<=0.195 ==> 품종2
5. 만약 X[0] >0.461 & X[6] >0.262 & X[4] >0.195 ==> 품종1