Supervised Learning

“**선형 모델**에서 **Weight & bias**”와 “**feature 수 변화**”를 중심으로

김 양 곤

발표 목차

1. Weight & bias
   1. 선형 모델
   2. 신경망
2. feature 줄이기, 늘리기
   1. Kernelized SVM
   2. 결정 트리, 결정 트리 앙상블

책의 목차

1. k-최근접 모델
2. 선형 모델
3. 나이브 베이즈 분류
4. 결정 트리
5. 결정 트리 앙상블
6. 커널 SVM
7. 신경망

1. 선형 모델 *for Regression*

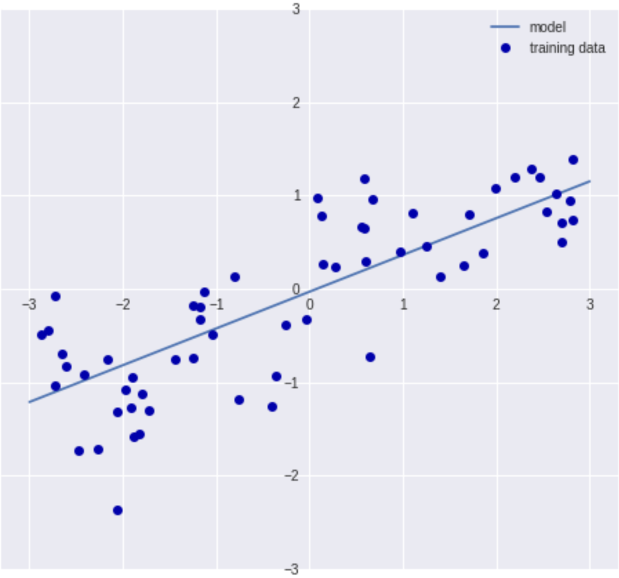
$$\hat{y} = w[0] \cdot x[0] + w[1] \cdot x[1] + \cdots + w[p] \cdot x[p] + b$$

w와 **b**는 학습 시킬 파라미터

y는 예측값

x[0] … x[p]는 한 데이터의 feature 값

ex)



$$y = w \cdot x + b$$

x축 : feature
y축 : target

feature가 한 개 일 때, 직선
두 개 일 때, 평면

1. 선형 모델 *for Regression*

W, b를 어떤 조건으로 학습시킬 것인가?

- *Least Squares* : MSE의 **최솟값**을 찾는 것

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

➡

$$\frac{\partial MSE}{\partial b} = \frac{1}{n}\frac{\partial \sum_{i=1}^{n}(y_i - b_i - wx_i)^2}{\partial b} = 0$$

$$\frac{\partial MSE}{\partial w} = \frac{1}{n}\frac{\partial \sum_{i=1}^{n}(y_i - b - w_i x_i)^2}{\partial w} = 0$$

➡

$$b = \hat{y} - w \cdot \hat{x}$$

$$w = \frac{\sum x_i (y_i - \hat{y})}{\sum x_i (x_i - \hat{x})}$$

- *Ridge* $MSE + \alpha \sum_{i=1}^{n} w_i^{\,2}$ 모든 특성이 출력에 주는 영향을 최소화. train data 성능 감소

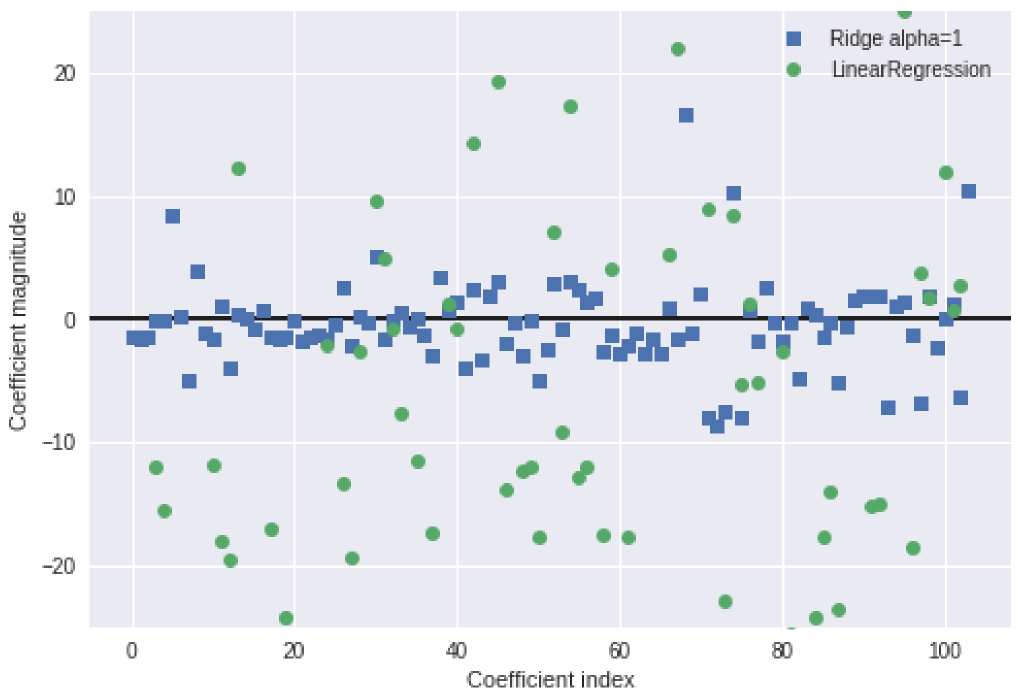
- *Lasso* $MSE + \alpha \sum_{i=1}^{n} | w_i |$ 어떤 특성은 출력에 영향을 아예 못 준다 (특정 w=0)

  feature selection : 모델에 가장 중요한 특성만 남긴다.

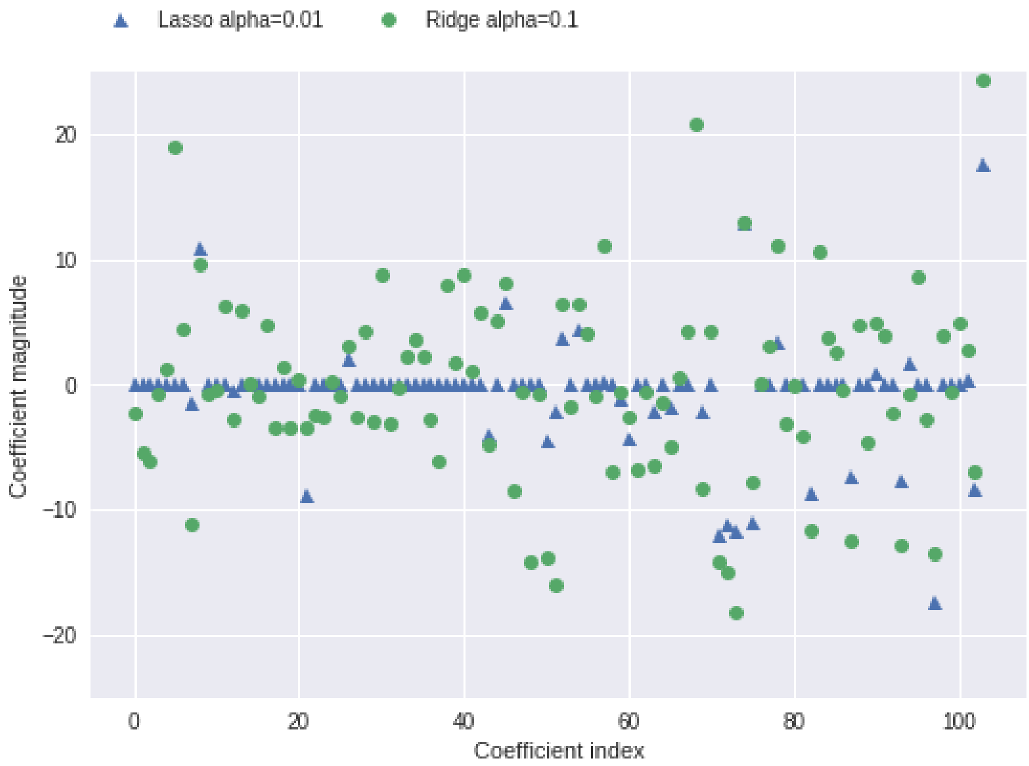
1. 선형 모델 *for Regression*

W, b를 어떤 조건으로 학습시킬 것인가?

Least Squares VS *Ridge*($\alpha = 1$)



Lasso ($\alpha = 0.01$) VS *Ridge* ($\alpha = 0.1$)



1. 선형 모델 *for Classification*

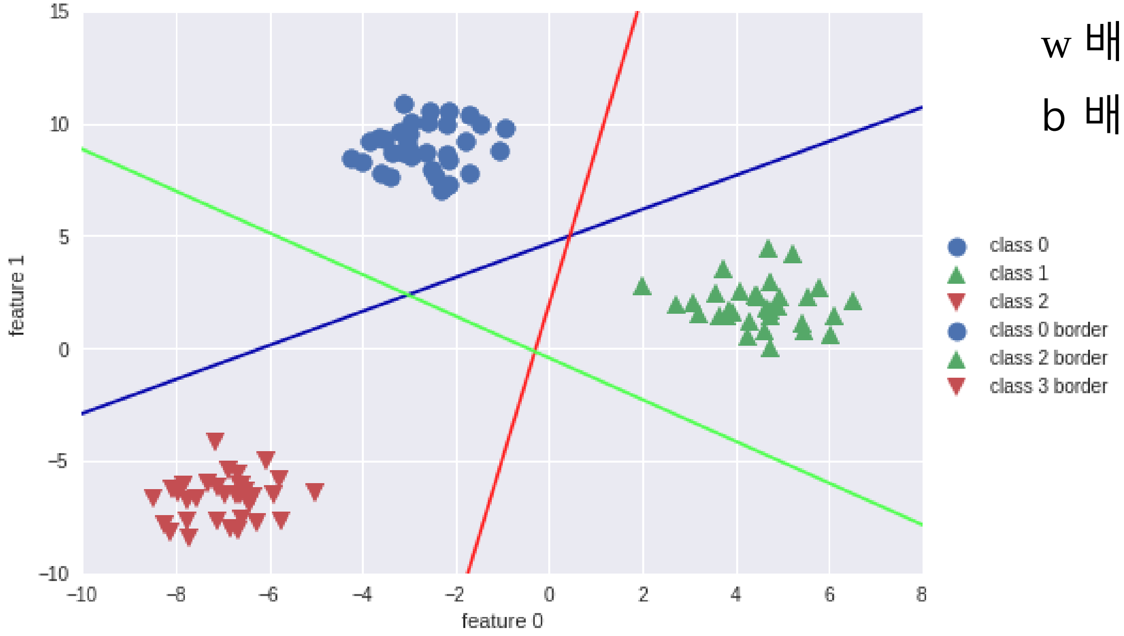
$$\hat{y} = w[0]\cdot x[0] + w[1]\cdot x[1] + \cdots + w[p]\cdot x[p] + b > 0$$

Regression과 달리 Classification에서는 y가 결정 경계

Regression과 달리 feature가 두 개 일 때, 직선

1. 선형 모델 *for Classification*

ex)



w 배열의 크기 : (3, 2) => $(w_{11}\ w_{12}), (w_{21}\ w_{22}), (w_{31}\ w_{32})$

b 배열의 크기 : (3,) => $(b_1), (b_2), (b_3)$

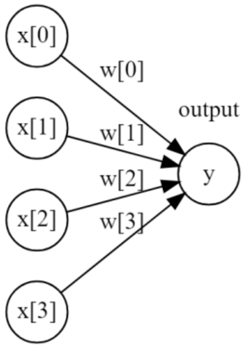
$$w_{11}x + w_{12}y + b_1 = 0$$

$$y = \frac{-(w_{11}x + b_1)}{w_{12}}$$

```
#세 개의 이진 분류기를 시각화
mglearn.discrete_scatter(X[:,0],X[:,1],y)
line = np.linspace(-15,15)
for coef, intercept, color in zip(linear_svm.coef_, linear_svm.intercept_,mglearn.cm3.colors):
  plt.plot(line,-(line*coef[0]+intercept)/coef[1],c=color)
```

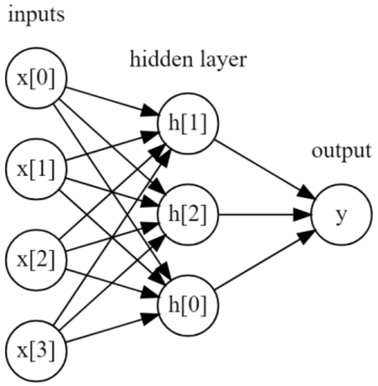
1. 선형 모델 *for Multilayer Perceptron*

- $\hat{y} = w[0]\cdot x[0] + w[1]\cdot x[1] + w[2]\cdot x[2] + w[3]\cdot x[3]$



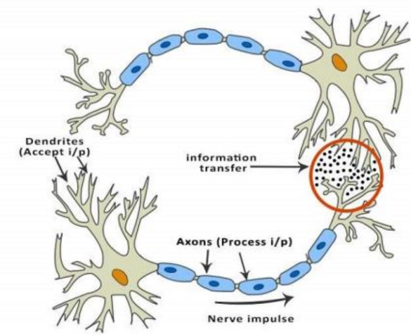
- $h = w[0]\cdot x[0] + w[1]\cdot x[1] + w[2]\cdot x[2] + w[3]\cdot x[3]$

  $\hat{y} = \hat{w}[0]\cdot h[0] + \hat{w}[1]\cdot h[1] + \hat{w}[2]\cdot h[2]$



모델의 예측값이 한번의 weighted sum 연산으로
결정하는 것이 아니라 여러 번에 걸쳐 반복한다.

뇌의 뉴런의 동작 방식을 모방한 것이다.



출처 : http://www.ijcstjournal.org/volume-5/issue-6/IJCST-V5I6P11.pdf

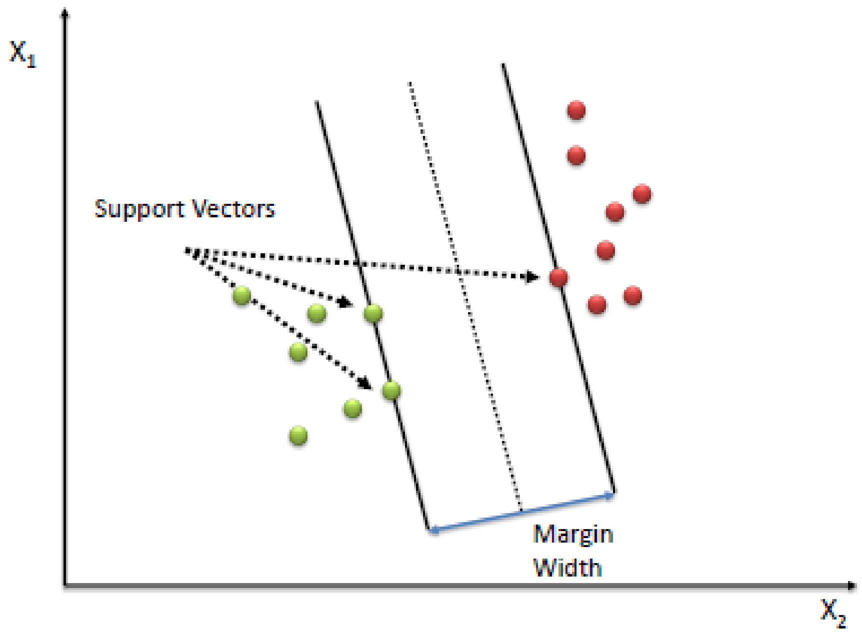
1. Weight & bias
    1. 선형 모델
    2. 신경망

2. feature 줄이기, 늘리기
    1. Kernelized SVM
    2. 결정 트리, 결정 트리 앙상블

2. feature 줄이기, 늘이기

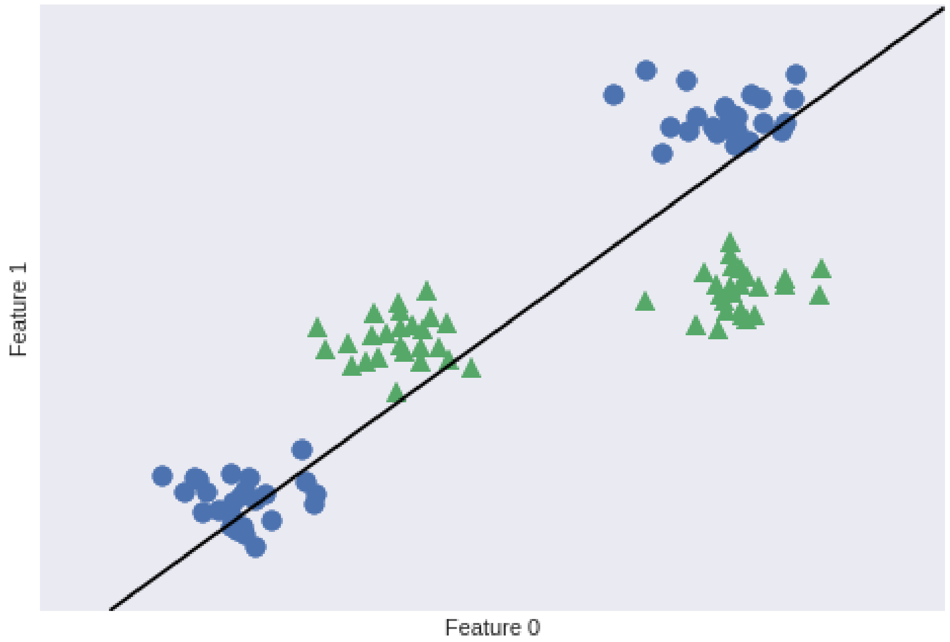
선형 Support Vector Machine

Margin을 크게 가지는 W, b를 찾는다.



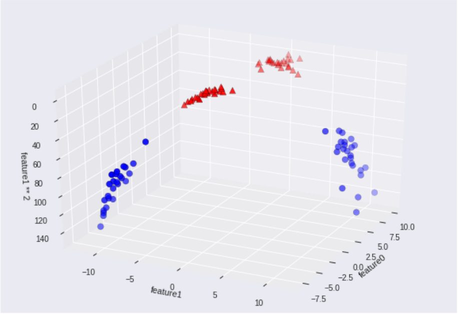
출처 : https://ratsgo.github.io/machine%20learning/2017/05/23/SVM/

선형 결정 경계를 긋기 힘들다



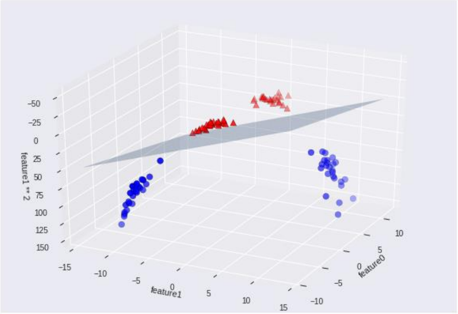
2. feature 줄이기, 늘이기

Kernelized Support Vector Machine



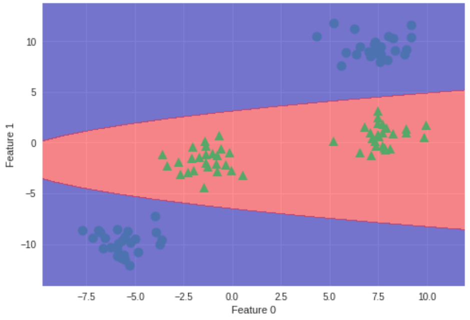
두번째 feature를 제곱하여 추가해주어
데이터가 마치 세개의 feature를 가진 것 처럼 표현한다

```
X_new = np.hstack([X, X[:, 1:] ** 2])
```



ax+by+cz+d=0 꼴의 선형 결정 경계를 그릴 수 있다

```
ZZ = (coef[0] * XX + coef[1] * YY + intercept) / -coef[2]
```



임시로 만들어 주었던 세번째 feature를 없애둔다
(위에서 구한 결정 경계를 xy 평면으로 사영시킨다)

2. feature 줄이기, 늘이기

Kernelized Support Vector Machine

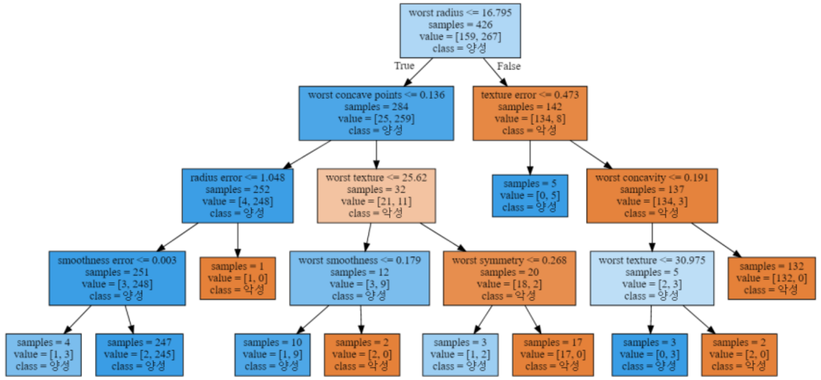
주어진 데이터의 특성을 이용하여 고차원 특성 공간으로 확장
고차원 특성 공간에서 두 범주를 구분하는 초평면을 찾는 것.

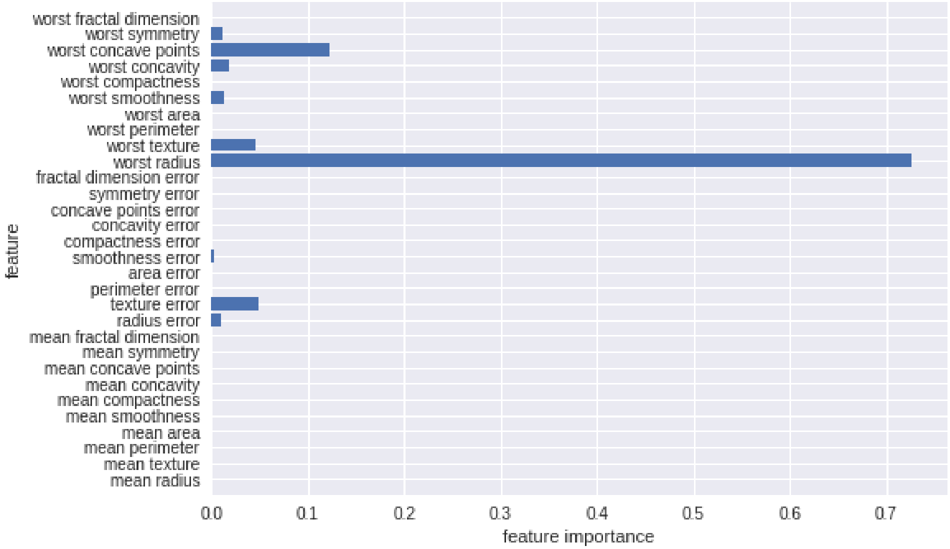
➡ feature을 늘여서 모델의 정확도를 높인다.

반대로, 모델의 정확도를 높이고 보니
데이터의 특성을 명확하게 규정하는 주요 특성만 남아 feature 수가 준 경우도 있다.

2. feature 줄이기, 늘이기

Decision Tree





- breast cancer data : (569, 30)

  Tree Classifier의 max depth를 4로 지정

  Test set accuracy가 0.951

- 30개의 feature 중 9개 만 사용

  "worst radius" feature가 가장 중요한 특성

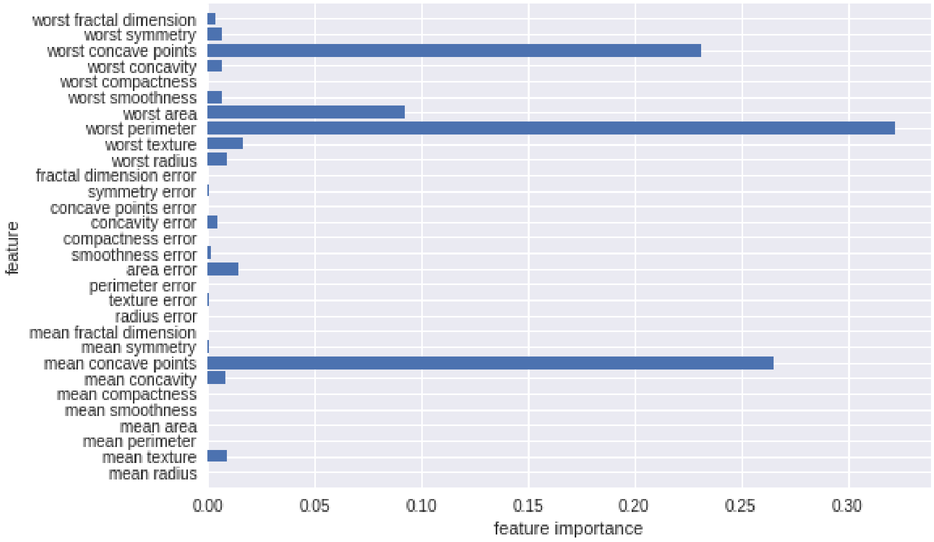
  (특성이 선택 되지 않았다고 해서 유용하지 않다고 단정 지을 수 없음. 다른 특정과 동일한 정보를 지니는데, 단지 선택 받지 못했을 수도 있다)

2. feature 줄이기, 늘이기

Gradient Boosting decision tress

Decision Tree의 단점은 Training Data에 과대 적합 되어 버림.
이 문제를 해결하고자 여러 모델을 연결하여 더 강력한 모델을 만드는 앙상블 기법 사용.

Gradient Boosting decision tress
이전 트리의 오차를 보완하는 방식으로 강력한 사전 가지 치기(learning_rate, n_estimators)



- Test set accuracy가 0.965

30개의 feature 중 18개 만 사용
“worst radius” feature를 비롯한 여러 특성 반영