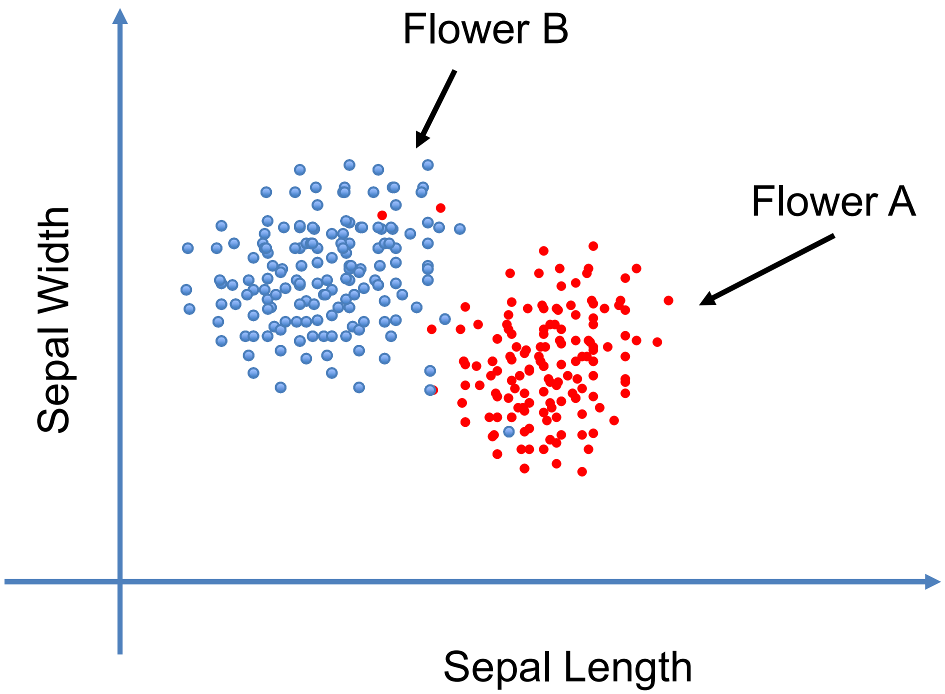
Classification-Logistic Regression

Week 4

What is Classification?



Which ones are classification problems?

☒ Predicting the price of a car (based on model, brand, year, etc.)

☑ Predicting if an email is spam or not

☑ Predicting if the price of a car is below $15K or not

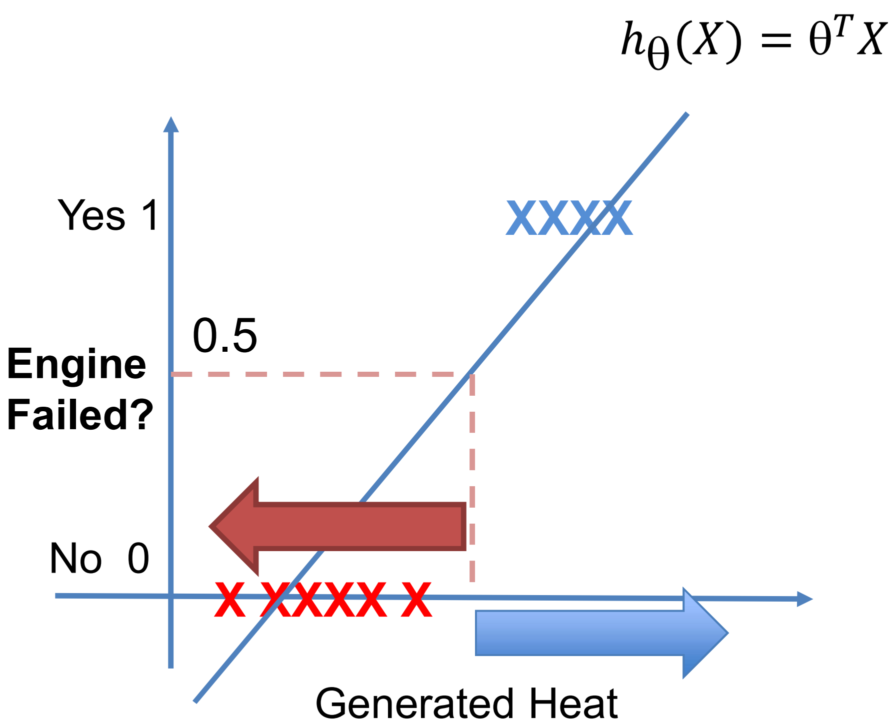
$\hat{y} \in \{0, 1\}$ → 0: "Negative Class"

→ 1: "Positive Class"

$\hat{y} \in \{0, 1, 2, 3, 4\}$

Multiclass Classification

Algorithm Development

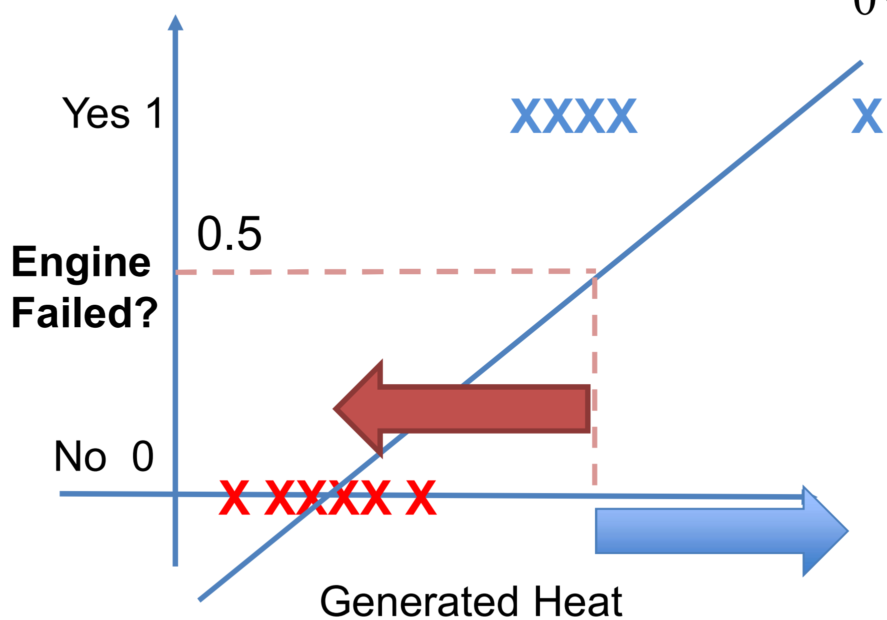


If $h_\theta(X)$ >= 0.5, then $\hat{y}$ =1

If $h_\theta(X) < 0.5$, then $\hat{y}$ =0

Algorithm Development

$$h_\theta(X) = \theta^T X$$



If $h_\theta(X) \geq 0.5$, then $\hat{y} = 1$

If $h_\theta(X) < 0.5$, then $\hat{y} = 0$

Algorithm Development

Another problem of using linear regression in classification,

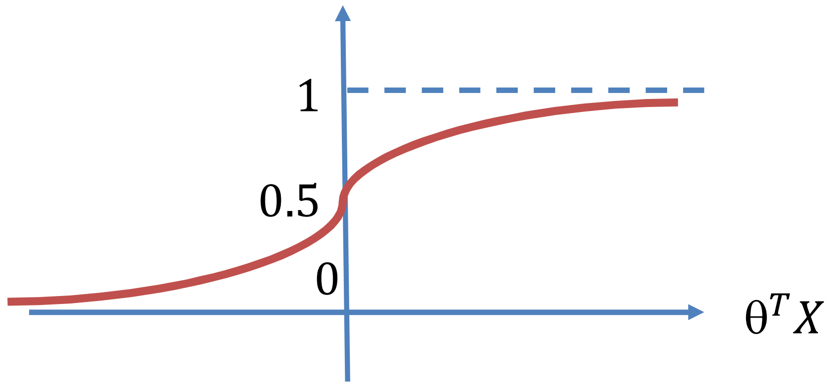
In classification problems, → $\hat{y} \in \{0, 1\}$

In linear regression problems, → $h_\theta(X) = \theta^T X$ can be larger than 1 or smaller than 0

In fact we want → $0 \leq h_\theta(X) \leq 1$

Logistic Regression Hypothesis

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$



Suppose we know θ^T and we get $h_\theta(X_{new}) = 0.8$ for the new engine.

Question: What is the meaning of $h_\theta(X_{new}) = 0.8$?

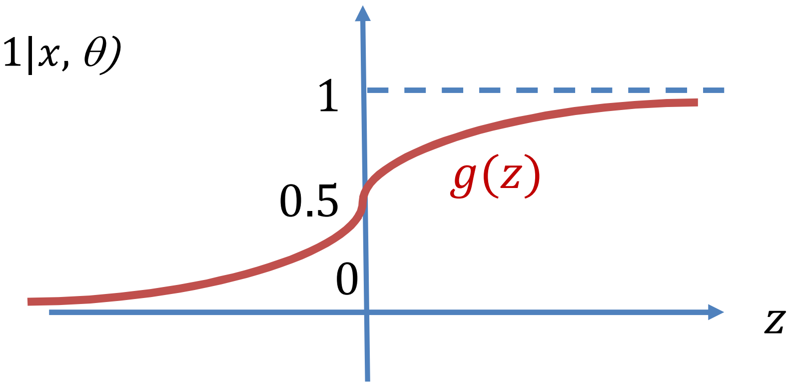
Answer: There is an 80% chance that new engine is broken!

Logistic Regression Hypothesis

$$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}} = P(y = 1|x, \theta)$$

$$h_\theta(X) = g(\theta^T X)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



Predict $y = 1$ ⟹ if $h_\theta(X) = g(\theta^T X) \geq 0.5$ ⟹ $\theta^T X = z \geq 0$

Predict $y = 0$ ⟹ if $h_\theta(X) = g(\theta^T X) < 0.5$ ⟹ $\theta^T X = z < 0$

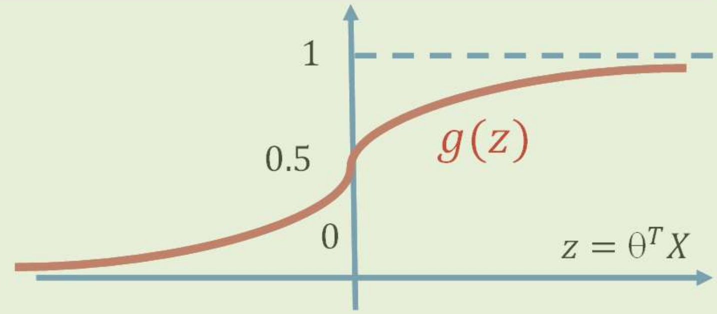
How this hypothesis makes predictions?

Decision Boundary

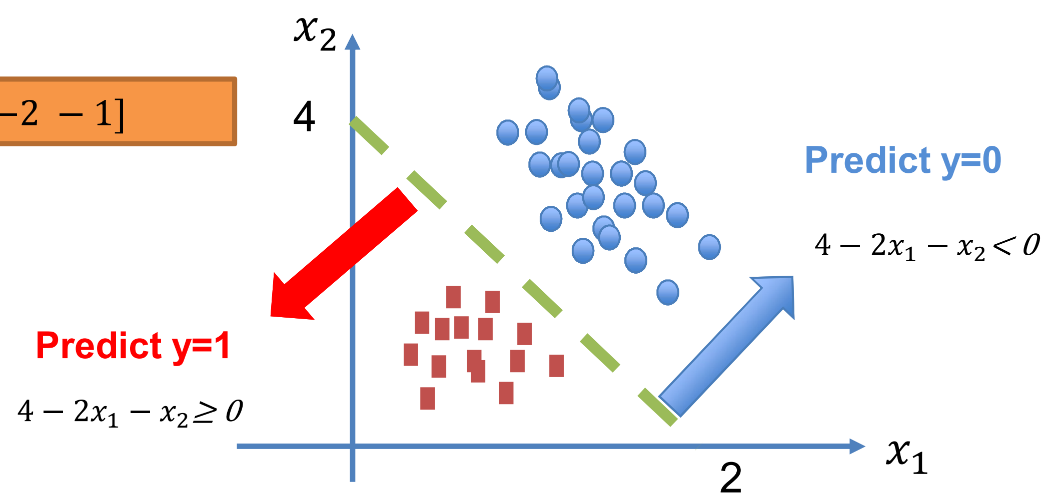
$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}} = P(y = 1|x, \theta)$

Predict $y = 1$ $\Rightarrow$ $\theta^T X = z \geq 0$

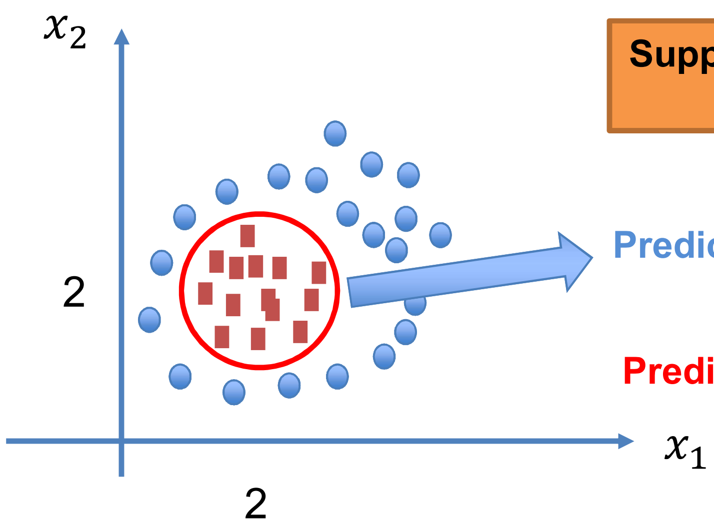
Predict $y = 0$ $\Rightarrow$ $\theta^T X = z < 0$



Suppose: $\theta^T = [4, -2 \ \ -1]$



Non-Linear Decision Boundary



Suppose:

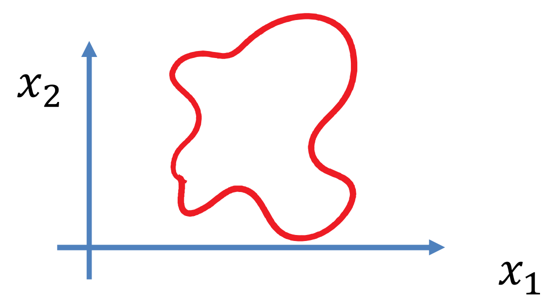
$$\theta^T X = -7 + 4x_1 + 4x_2 - {x_1}^2 - {x_2}^2$$

Predict y=0 if: $-7 + 4x_1 + 4x_2 - {x_1}^2 - {x_2}^2 < 0$

Predict y=1 $-7 + 4x_1 + 4x_2 - {x_1}^2 - {x_2}^2 \geq 0$

What if we have even more non-linear terms?

$$h(\theta^T X) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 {x_1}^2 + \theta_4 {x_1}^2 x_2 + \theta_5 {x_1}^2 {x_2}^2 + \cdots)$$



Fitting $\theta_0, \theta_1, \theta_2, \ldots$

What we have:

Training set:

$$X=\begin{bmatrix} x_1^{(1)} & \ldots & x_1^{(n)} \\ x_2^{(1)} & \ldots & x_1^{(n)} \\ \vdots & \ldots & \vdots \\ x_m^{(1)} & \ldots & x_m^{(n)} \end{bmatrix}$$ and $$y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$ where $y_i \in \{0,1\}$

What we want:

Find $\theta^T = [\theta_0, \theta_1, \theta_2, \ldots]$ in $$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}}$$

Cost Function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

Linear regression

$$h_\theta(X) = \theta^T X$$

$J(\theta)$ Convex

Logistic regression

$$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}}$$

$J(\theta)$ Non-convex

Cost Function:

Different cost function which is convex:

$$\boxed{J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{-y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - (1-y^{(i)})\ \log\left(1-h_\theta\left(x^{(i)}\right)\right)}}$$

$$cost(h_\theta(x), y)$$

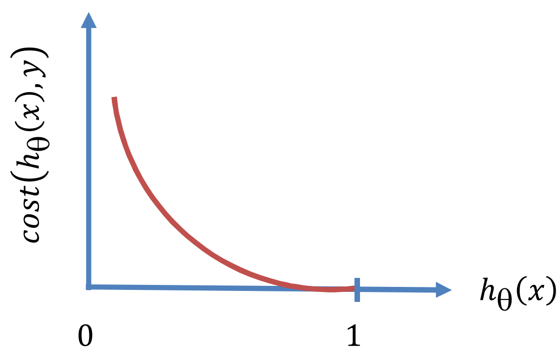
$$\boxed{J(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost(h_\theta(x), y)}$$

Cost Function:

$$\boxed{cost\left(h_\theta(x), y\right) = -y^{(i)} \log\left(h_\theta(x)\right) - (1 - y^{(i)}) \log\left(1 - h_\theta(x)\right)}$$

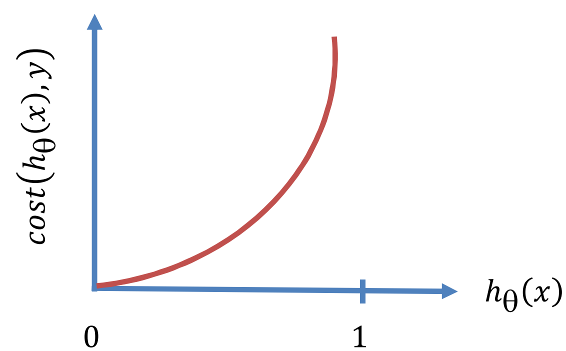
When y=1:

$$cost\left(h_\theta(x), y\right) = -\log\left(h_\theta(x)\right)$$



When y=0:

$$cost\left(h_\theta(x), y\right) = -\log\left(1 - h_\theta(x)\right)$$



Gradient Descent:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\ \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

To get, $\min_\theta\ J(\theta)$: $\longrightarrow$

Repeat, {

$$\theta_k := \theta_k - \alpha \frac{\partial}{\partial \theta_k} J(\theta)$$

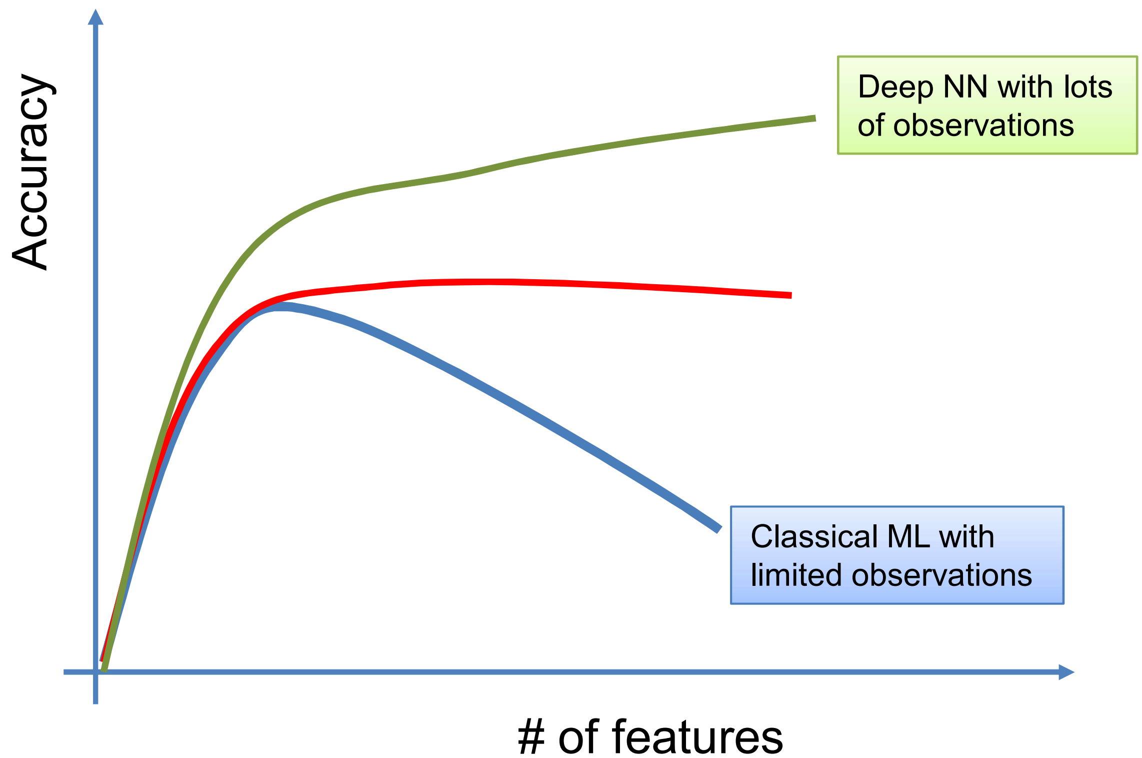
}

Repeat, {

$$\theta_k := \theta_k - \alpha \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_k^{(i)}$$

}

Feature Selection



Recursive Feature Elimination (RFE)

X1
X2
X3
X4

$$h_\theta(X) = 5 + 3X1 + 8X2 + \mathbf{0.1X3} - 5X4$$

X1
X2
X4

$$h_\theta(X) = 6 + \mathbf{2X1} + 11X2 - 6X4$$

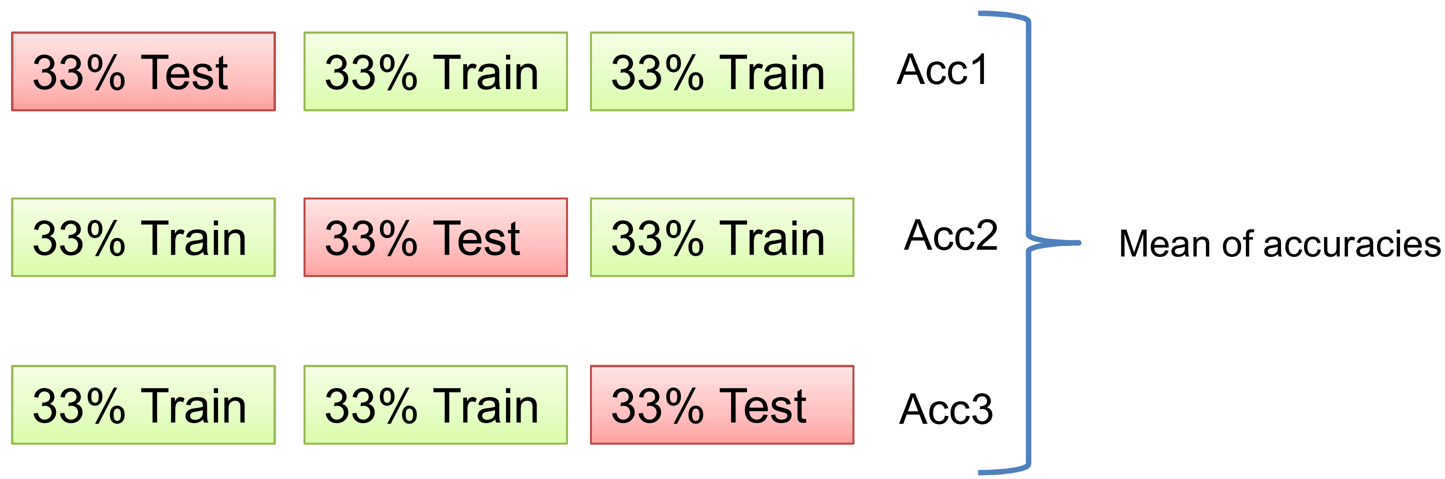
X2
X4

So what is the optimal number of features?

Kfold Cross Validation

Divide the whole data set to k buckets and select each bucket as test set and the rest as training sets.

Example,   K=3



RFE in Sklearn

The least important features are pruned from current set of features recursively.

sklearn.feature_selection.RFE(estimator, step=1, n_features_to_select=None, cv=3)

Example:

```
estimator = SVR(kernel="linear")
selector = RFE(estimator, step=1, )
selector = selector.fit(X, y)
```

```
selector. ranking_
array([1, 1, 1, 1, 1, 4, 2, 1, 1, 3])
```

RFE with Kfold Cross Validation

- Ranks the features with RFE.
- Cross-validates selection of the best number of features (K-Fold).

sklearn.feature_selection.RFECV(estimator, step=1, min_features_to_select=1, cv=3)

Example:

```
estimator = SVR(kernel="linear")
selector = RFECV(estimator, step=1, cv=5)
selector = selector.fit(X, y)
```

```
selector. ranking_
array([1, 1, 1, 1, 1, 4, 2, 1, 1, 3])
```

RFE in Sklearn

```
from sklearn import linear_model
estimator = linear_model.LinearRegression()
```

sklearn.feature_selection.RFE(estimator, n_features_to_select=None)

For other feature selection methods, refer to:

https://scikit-learn.org/stable/modules/feature_selection.html