

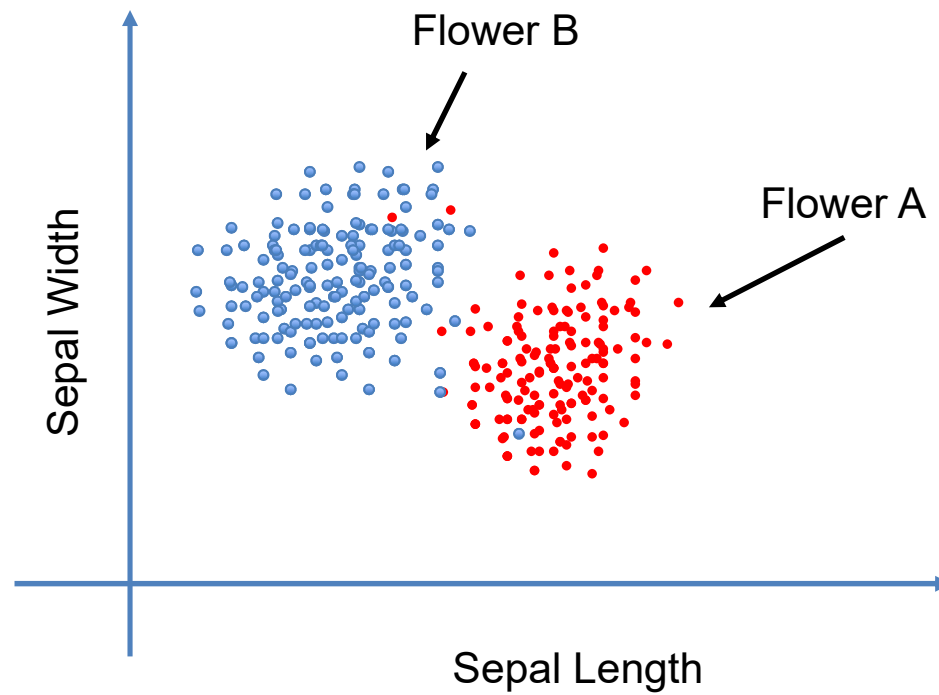
Classification-Logistic Regression

Week 4

Mohammad Esmalifalak

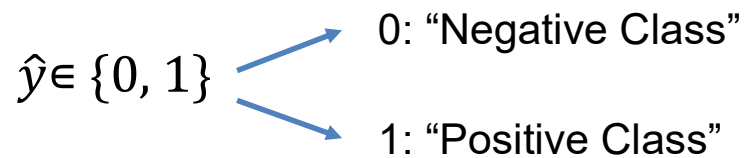


What is Classification?



Which ones are classification problems?

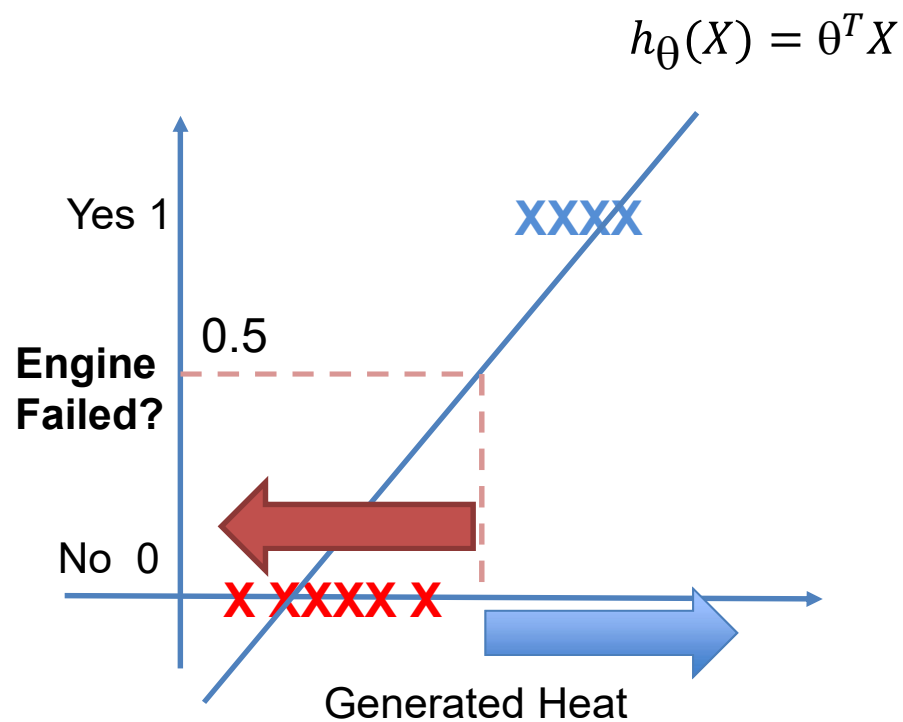
- ☒ Predicting the price of a car (based on model, brand, year, etc.)
- ☒ Predicting if an email is spam or not
- ☒ Predicting if the price of a car is below \$15K or not



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

Multiclass Classification

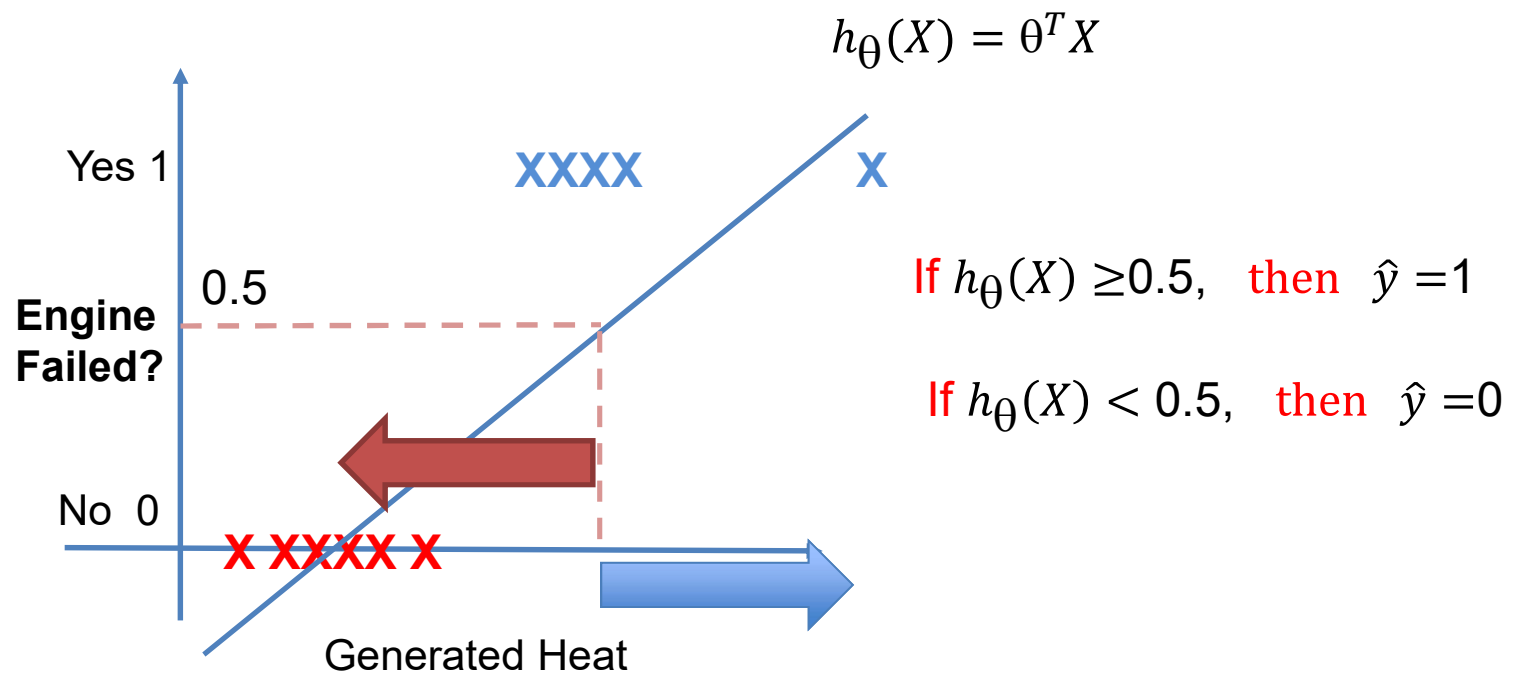
Algorithm Development



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

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

Algorithm Development



Algorithm Development

Another problem of using linear regression in classification,

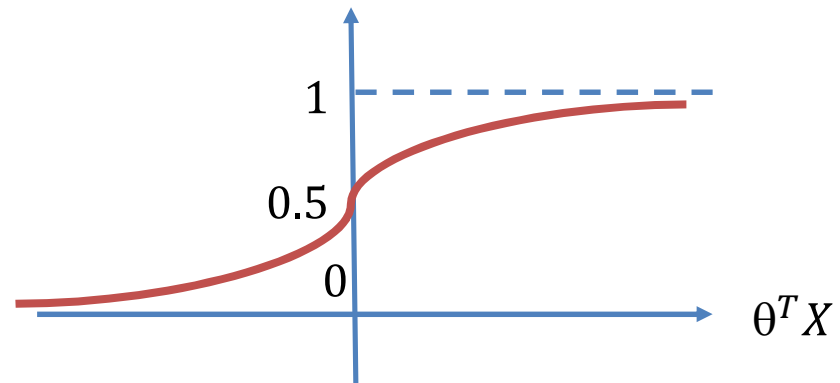
In classification problems, $\longrightarrow \hat{y} \in \{0, 1\}$

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

In fact we want $\longrightarrow 0 \leq h_{\theta}(X) \leq 1$

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 $h_{\theta}(X_{new}) = 0.8$?

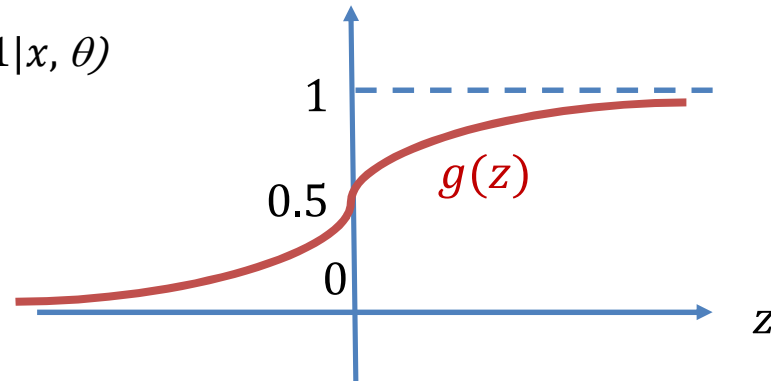
Answer: There is an 80% chance that new engine is broken!

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 $y = 1$ \longrightarrow if $h_{\theta}(X) = g(\theta^T X) \geq 0.5$ \longrightarrow $\theta^T X = z \geq 0$

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

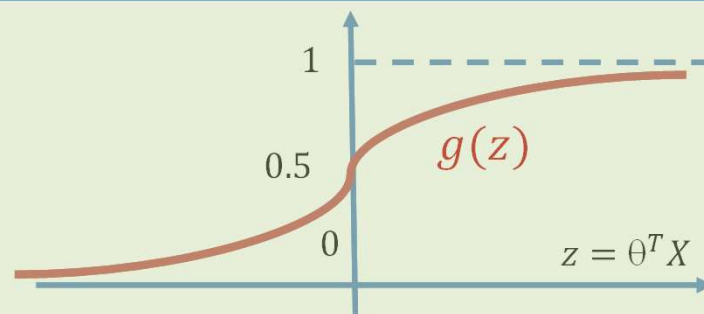
How this hypothesis makes predictions?

Decision Boundary

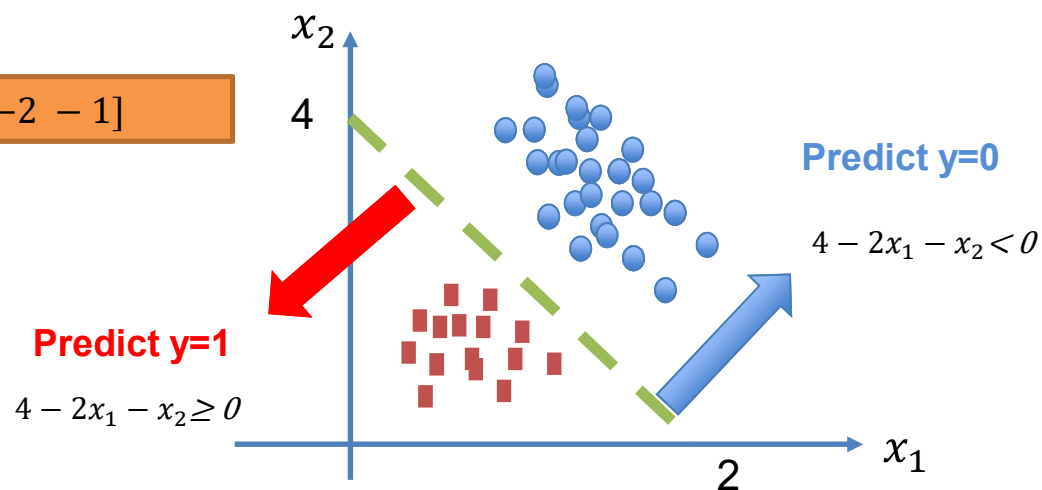
$$h_{\theta}(X) = \frac{1}{1+e^{-\theta^T X}} = P(y = 1|x, \theta)$$

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

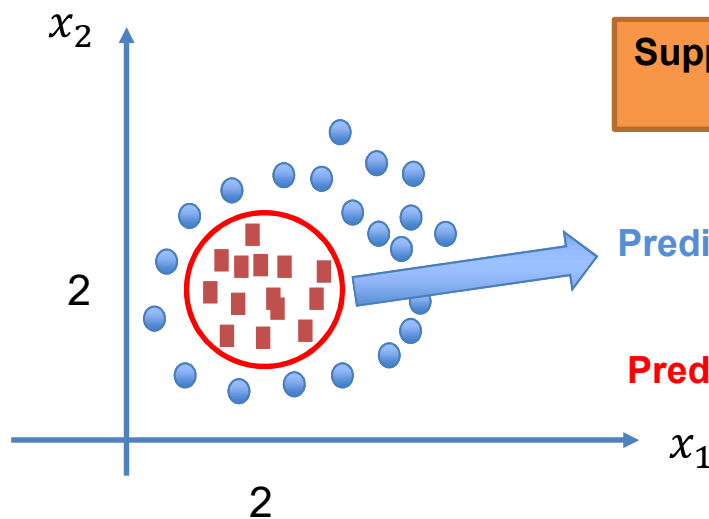
Predict $y = 0$ $\Rightarrow \theta^T X = z < 0$



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



Non-Linear Decision Boundary



Suppose:

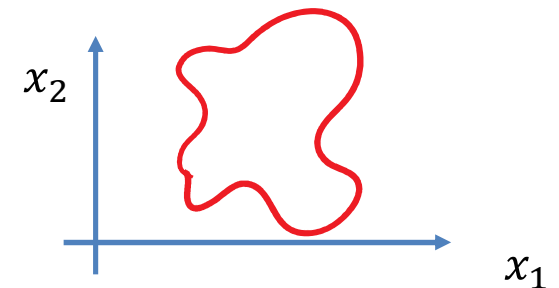
$$\theta^T X = -7 + 4x_1 + 4x_2 - x_1^2 - x_2^2$$

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?

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



Fitting $\theta_0, \theta_1, \theta_2, \dots$

What we have:

Training set:

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

What we want:

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

Cost Function:

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

The diagram shows the cost function $J(\theta) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$ at the top. A blue circle highlights the term $h_{\theta}(x^{(i)})$. Two red arrows point from this circle to the text 'Linear regression' on the left and 'Logistic regression' on the right. Below 'Linear regression' is the equation $h_{\theta}(X) = \theta^T X$, with a red arrow pointing down to the text ' $J(\theta)$ Convex'. Below 'Logistic regression' is the equation $h_{\theta}(X) = \frac{1}{1 + e^{-\theta^T X}}$, with a red arrow pointing down to the text ' $J(\theta)$ Non-convex'.

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:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \underbrace{-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))}_{\text{cost}(h_{\theta}(x), y)}$$

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

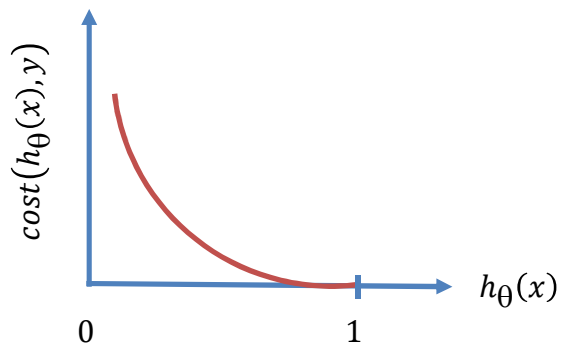
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{cost}(h_{\theta}(x), y)$$

Cost Function:

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

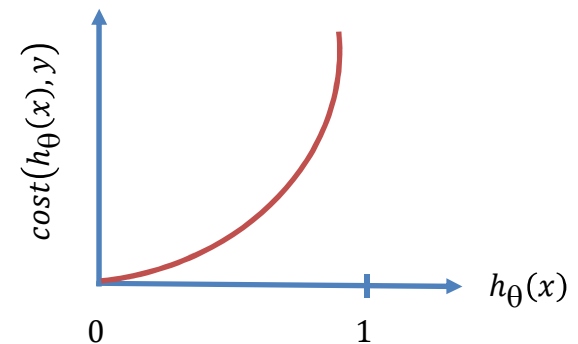
When $y=1$:

$$\text{cost}(h_{\theta}(x), y) = -\log(h_{\theta}(x))$$



When $y=0$:

$$\text{cost}(h_{\theta}(x), y) = -\log(1 - h_{\theta}(x))$$



Gradient Descent:

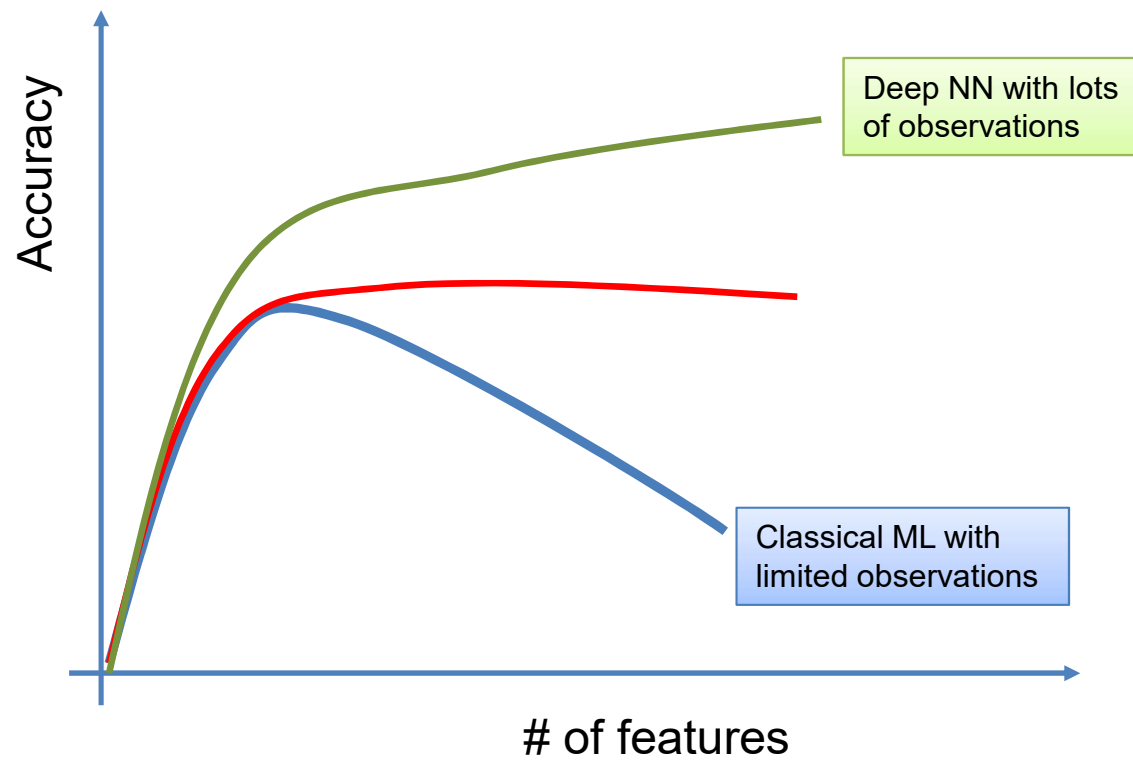
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m -y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))$$

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

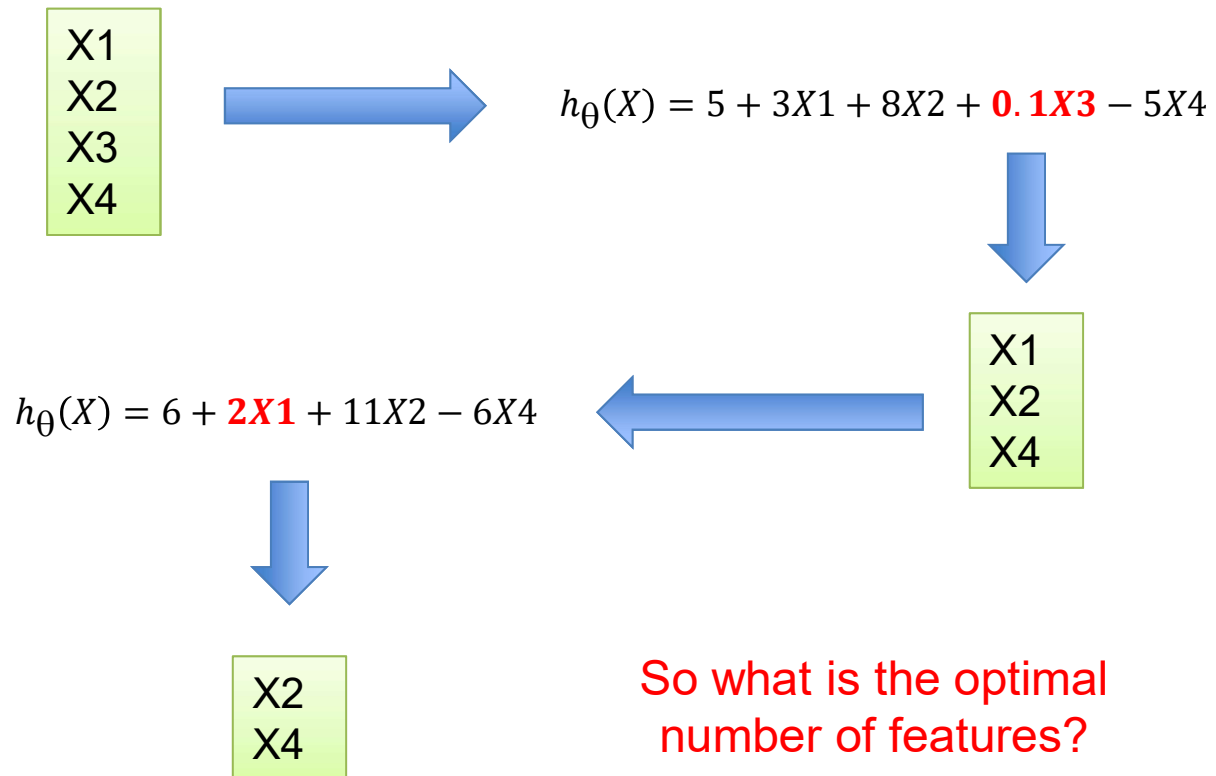
Repeat, {
 $\theta_k := \theta_k - \alpha \frac{\partial}{\partial \theta_k} J(\theta)$
}

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

Feature Selection



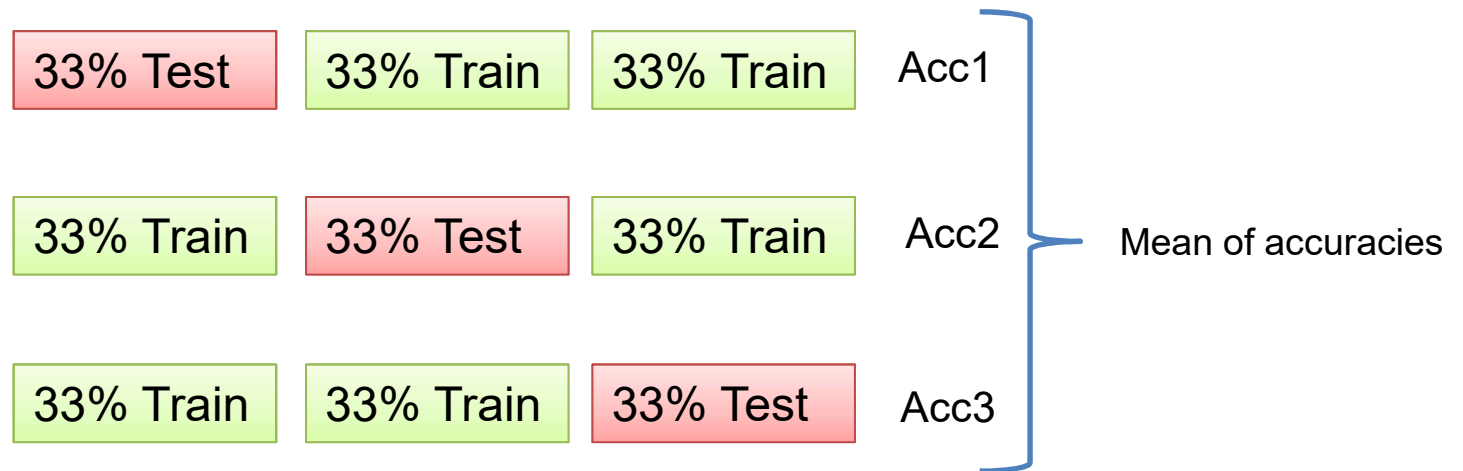
Recursive Feature Elimination (RFE)



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

