Machine Learning

Week 5 - Review - Jan 2020

About the course

Announcements

- This is our final session
- You can download this week’s Notebook folder if you haven’t and start Jupyter Notebook
- If you can, camera on is appreciated

Previously on ML..

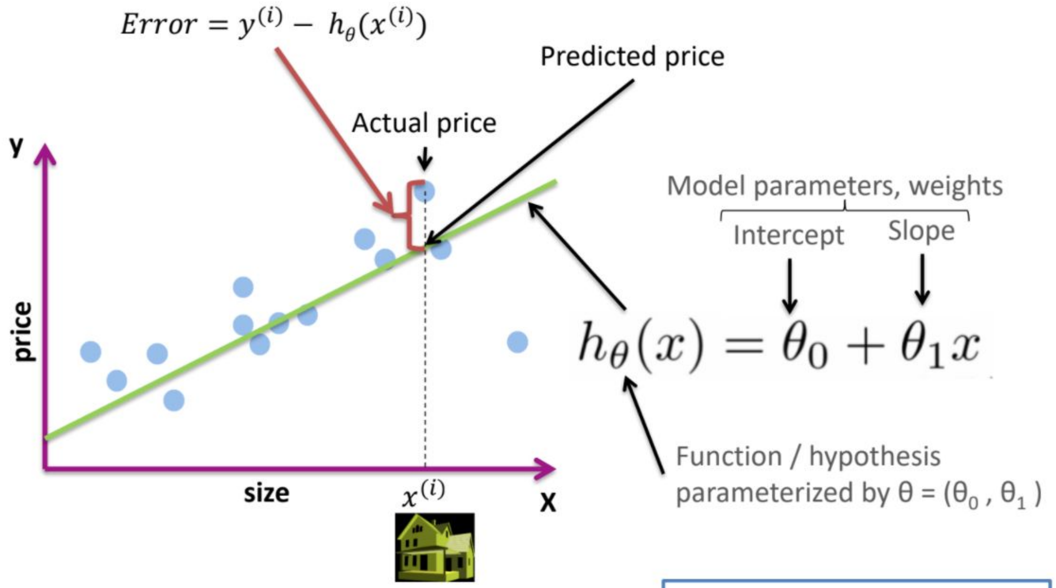
Linear Regression
(Record this course)

Linear Regression

Linear Regression is a statistical model used to predict the relationship **between independent and dependent variables**. In linear regression, the relationships are modeled using **linear predictor functions** whose unknown model parameters are **estimated** from the data.

Linear Regression

We start with a function

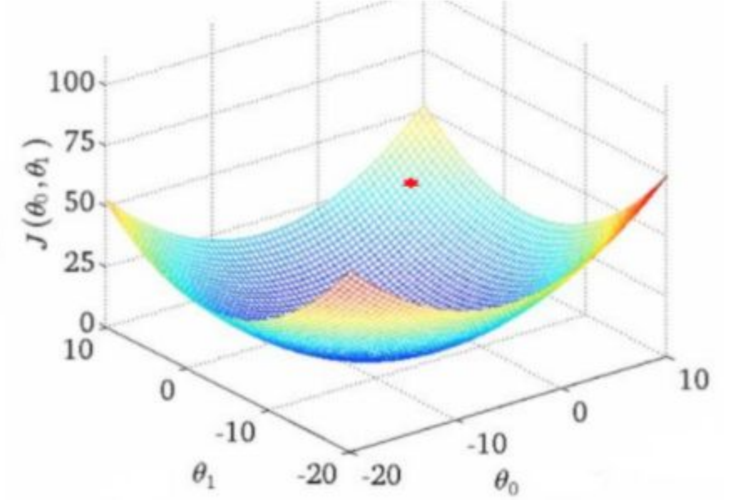


$$\boxed{y = \theta_0 + \theta_1 x + \varepsilon}$$

Linear Regression

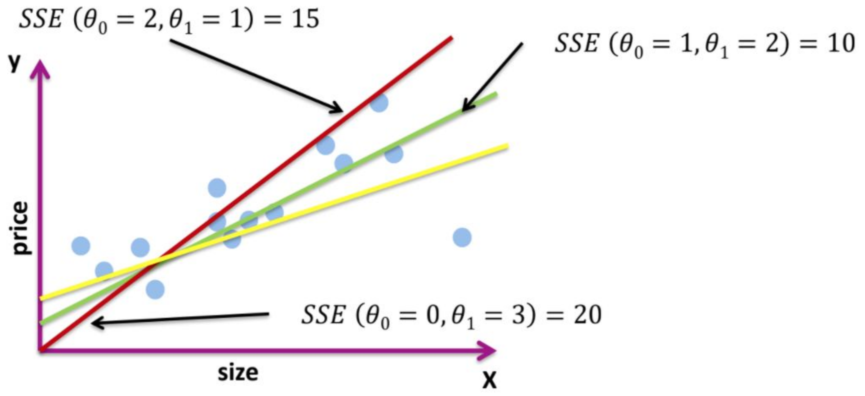
We then iteratively make predictions, apply the cost function and update parameters

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$



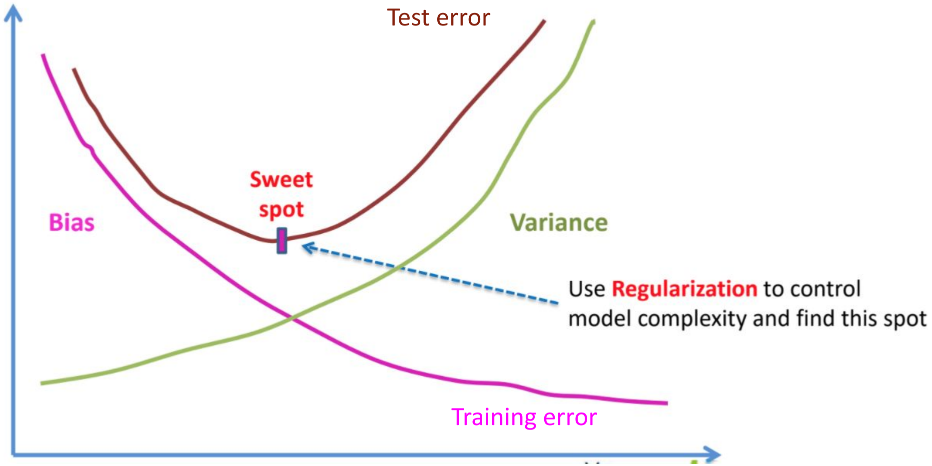
Linear Regression

The product of the learning process are the model's parameters



Linear Regression

Watch out for the overfitting!!



Linear Regression

Regularization can be helpful

- Ridge Regression (L_2 regularization)

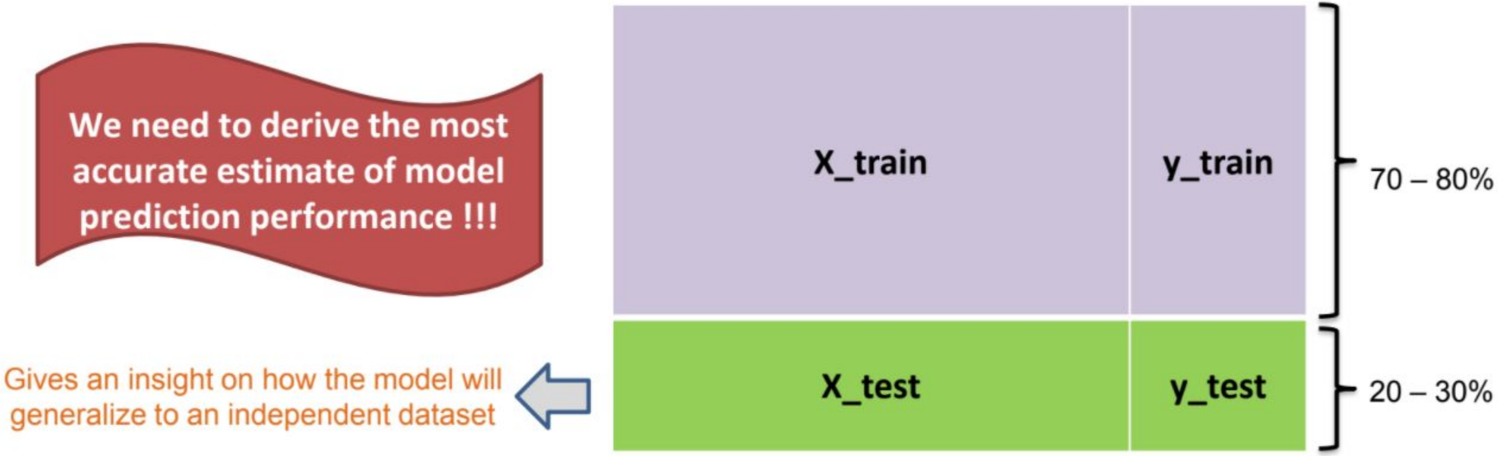
$$J(\theta) = \frac{1}{2m} \left[\sum_{i=0}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} (\theta_j)^2 \right]$$

- Lasso Regression (L_1 regularization)

$$J(\theta) = \frac{1}{2m} \left[\sum_{i=0}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} | \theta_j | \right]$$

Testing your system

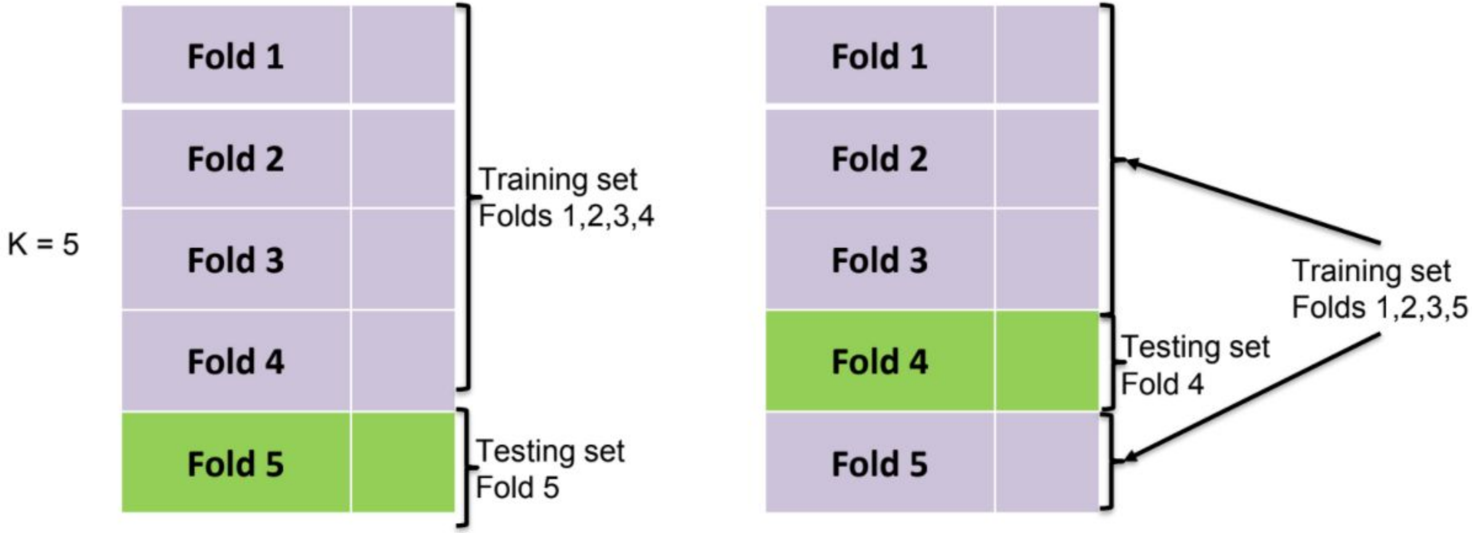
Always separate your dataset in train and test sets



Testing your system

Make even more separations!

- Partition the dataset into **K** folds (bins) of equal size.



Logistic Regression

Logistic Regression

Logistic regression is a statistical **model** that in its basic form uses a **logistic** function to **model** a binary dependent variable. In regression analysis, logistic regression (or logit regression) is estimating the **parameters** of a logistic model (a form of binary regression).

Logistic Regression

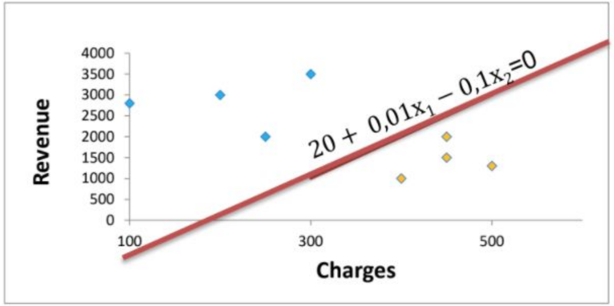
We start with a function - the Decision Boundary

Decision Boundary

- Example: Loan demand evaluation model
  - Predict the loan safety class given the revenue and the charges values
  - $h_\theta(x) = g\,(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = g\,(20 + 0{,}01\ \#revenue - 0{,}1\#charges)$

- Predict **1** : if $g(\theta^T x) \geq 0{,}5$ ➔ $\theta^T x \geq 0$
  - if $g\,(20 + 0{,}01x_1 - 0{,}1x_2) \geq 0.5$
  - $\Rightarrow 20 + \ 0{,}01x_1 - 0{,}1x_2 \geq 0$

- Predict **0** : if $g(\theta^T x) < 0{,}5$ ➔ $\theta^T x < 0$
  - if $g\,(20 + 0{,}01x_1 - 0{,}1x_2) < 0.5$
  - $\Rightarrow 20 + \ 0{,}01x_1 - 0{,}1x_2 < 0$



- $20 + \ 0{,}01x_1 - 0{,}1x_2{=}0$ is our decision boundary

Logistic Regression

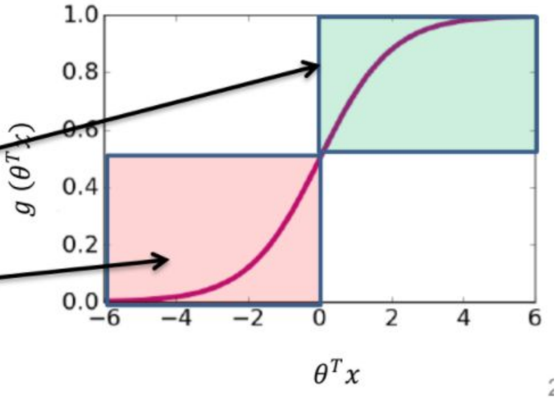
We then fit this decision boundary into the Logistic (Sigmoid function)

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

- $h_\theta(x)$ = **estimated probability that y = 1** given the input x parameterized by θ
  - Example: $h_\theta(x)$= 0,8 ➔ The probability that the loan is safe (y=1) is equal to 80%

- $h_\theta(x) = \mathrm{P}(\mathrm{y} = 1|\ \mathrm{x}; \theta) = 1 - P(\mathrm{y} = 0|\ \mathrm{x}; \theta)$
- $\mathrm{P}(\mathrm{y} = 1|\ \mathrm{x}; \theta) + P(\mathrm{y} = 0|\ \mathrm{x}; \theta) = 1$

- Hypothesis:
  - ❑ $y = 1\ when\ g(\theta^T x) \geq 0{,}5$ ➔ $\theta^T x \geq 0$
  - ❑ $y = 0\ when\ g(\theta^T x) < 0{,}5$ ➔ $\theta^T x < 0$

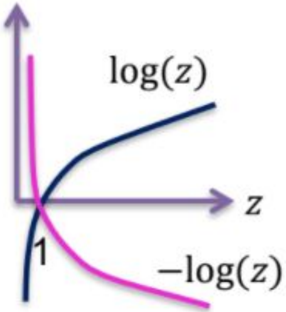


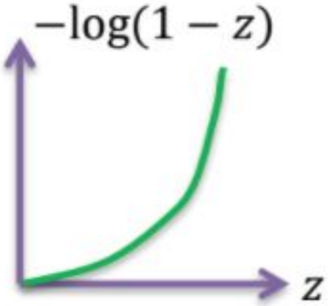
Logistic Regression

We then iteratively make predictions, apply the cost function and update parameters

- $\mathrm{Cost}(h_\theta(x), y) = \boxed{-(y) *\log(h_\theta(x))}\ \boxed{-(1-y)*\log(1-h_\theta(x))}$

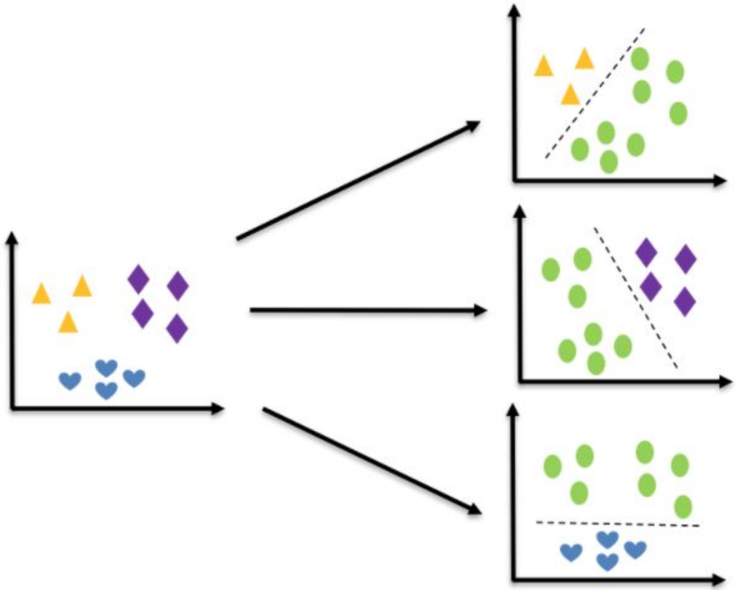
Predicted value   Real value





Logistic Regression - Multiclass

When predicting more than 2 classes we train a model for each class



$h_\theta^{1}(x)$: P(y = ▲| x; θ) ⇒ $\boxed{0.93}$

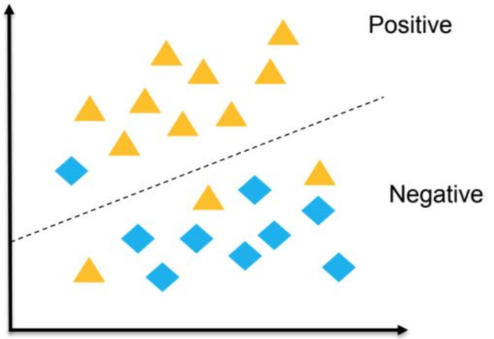
$h_\theta^{2}(x)$: P(y = ◆| x; θ) ⇒ 0.63

$h_\theta^{3}(x)$: P(y = ♥| x; θ) ⇒ 0.33

Logistic Regression - Performance

90% accuracy doesn’t tell the whole story

		Predicted class	
		Negative	Positive
Actual class	Negative	True Negative	False Positive
	Positive	False Negative	True Positive



$$\text{Precision} = \frac{\textit{True positive}}{\textit{True positive} + \textit{False positive}} = \frac{9}{9+1} = 90\%$$

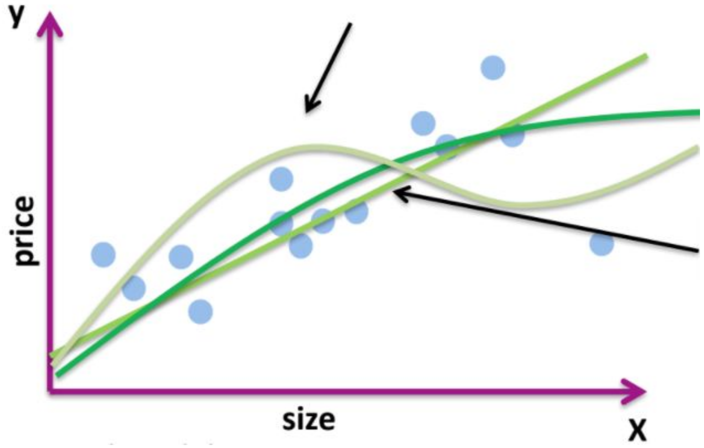
$$\text{Recall} = \frac{\textit{True positive}}{\textit{True positive} + \textit{False negative}} = \frac{9}{9+3} = 75\%$$

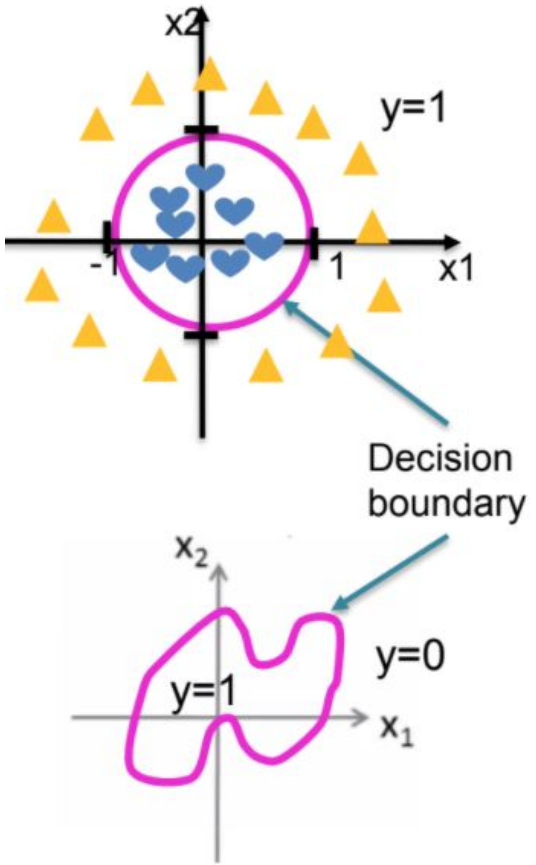
Polynomial Regression

Polynomial Regression

Useful for both continuous and discrete predictions

$$h_\theta(x) = g\,(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 {x_1}^2 + \theta_4 {x_2}^2 + \theta_5 {x_1}^3 + \theta_6 {x_2}^3)$$





Decision Trees

Conclusion