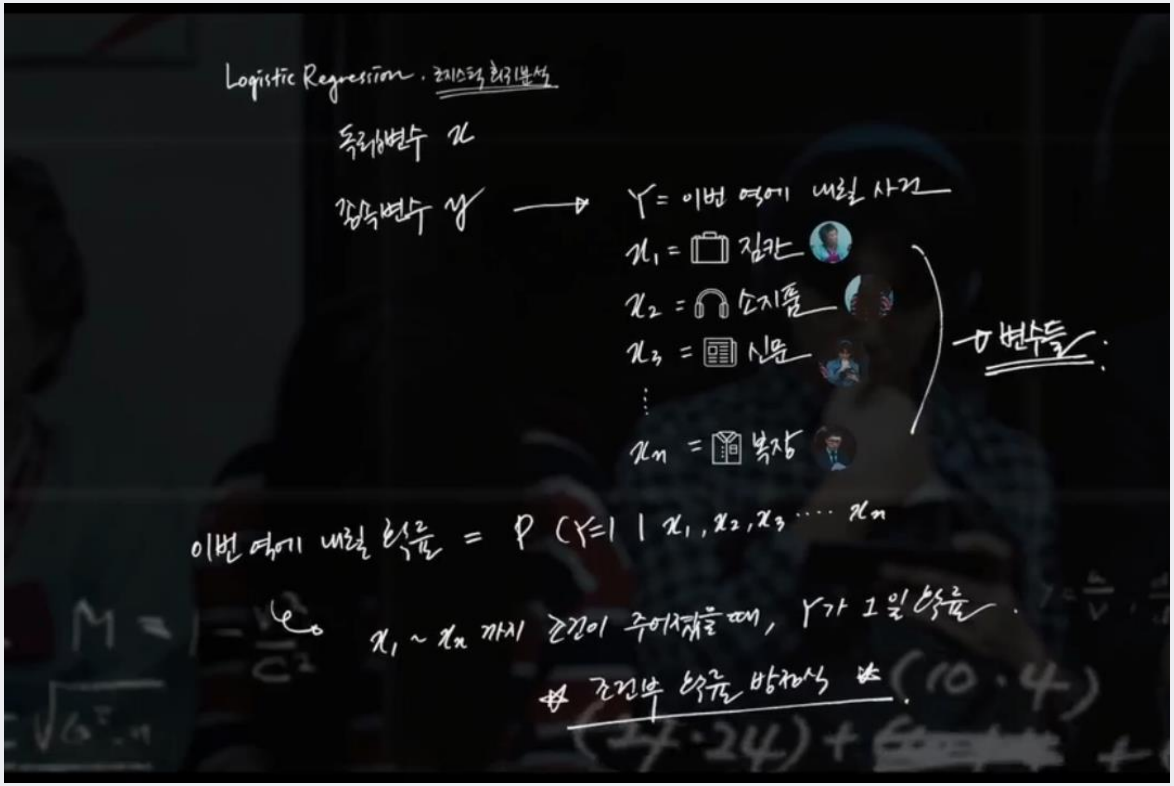


Lecture 6: Logistic Regression

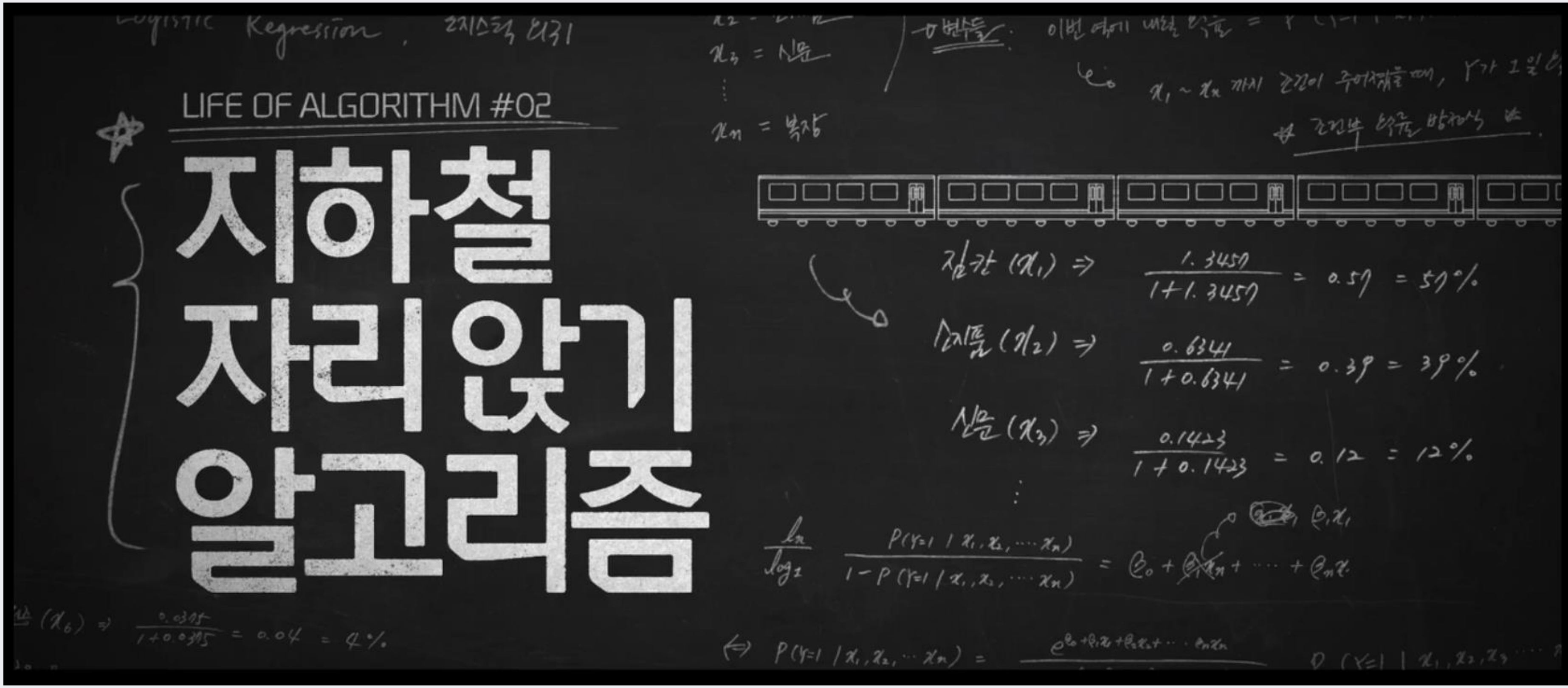
Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

Logistic Regression: Intro.



http://channel.hyundaicard.com/v/dh0005

Logistic Regression

- Classification

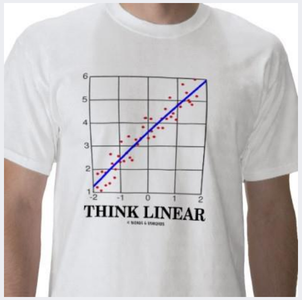
Men

Vs.

Women

Revisit Multiple Linear Regression

- Goal
  - ✓ Fit a linear relationship between a quantitative dependent variable Y and a set of predictors $X_1, X_2, \ldots, X_p$.

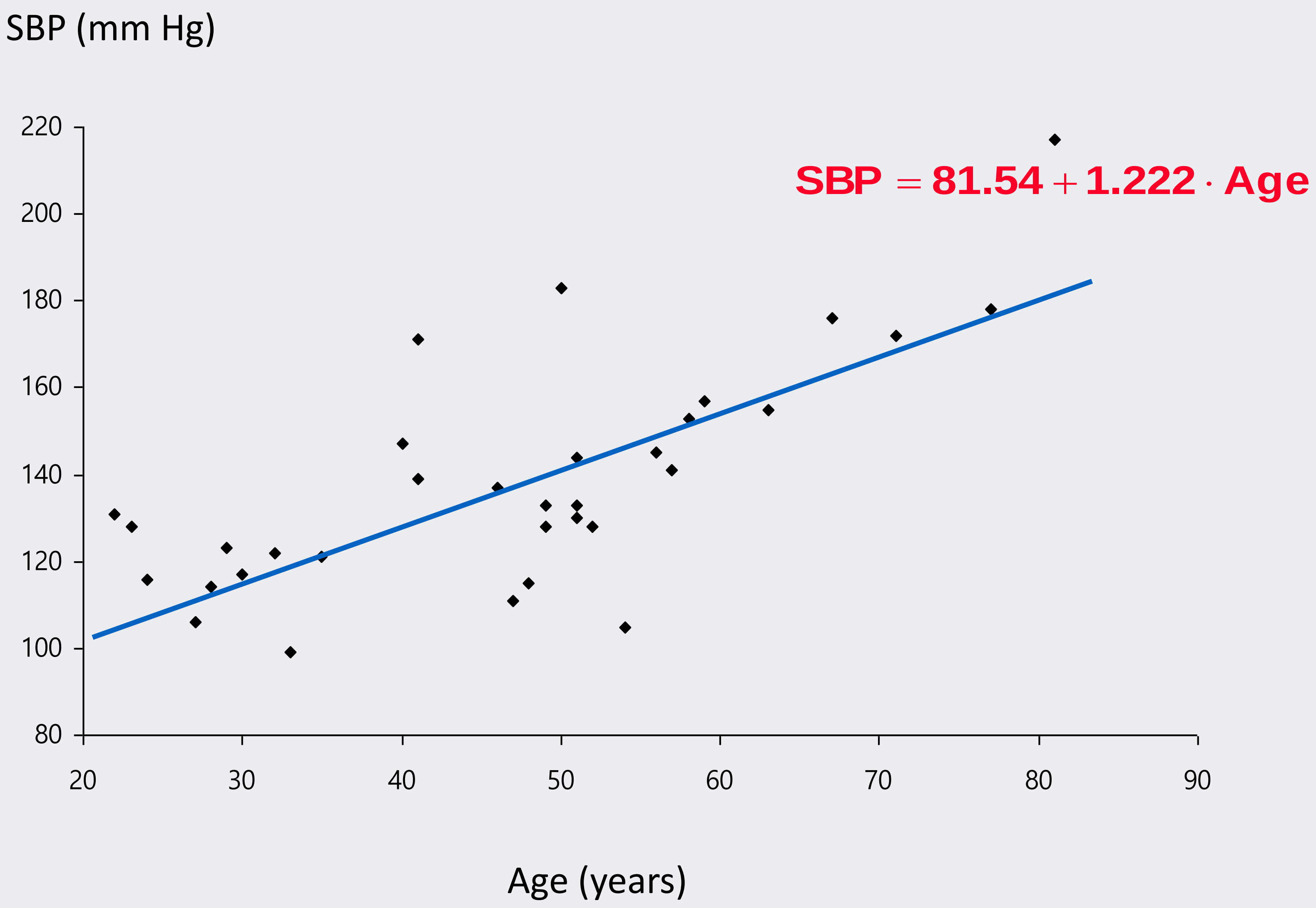


$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

- Example 1
  - ✓ Age and systolic blood pressure (SBP) among 33 adult women.

Age	SBP	Age	SBP	Age	SBP
22	131	41	139	52	128
23	128	41	171	54	105
24	116	46	137	56	145
27	106	47	111	57	141
28	114	48	115	58	153
29	123	49	133	59	157
30	117	49	128	63	155
32	122	50	183	67	176
33	99	51	130	71	172
35	121	51	133	77	178
40	147	51	144	81	217

Revisit Multiple Linear Regression



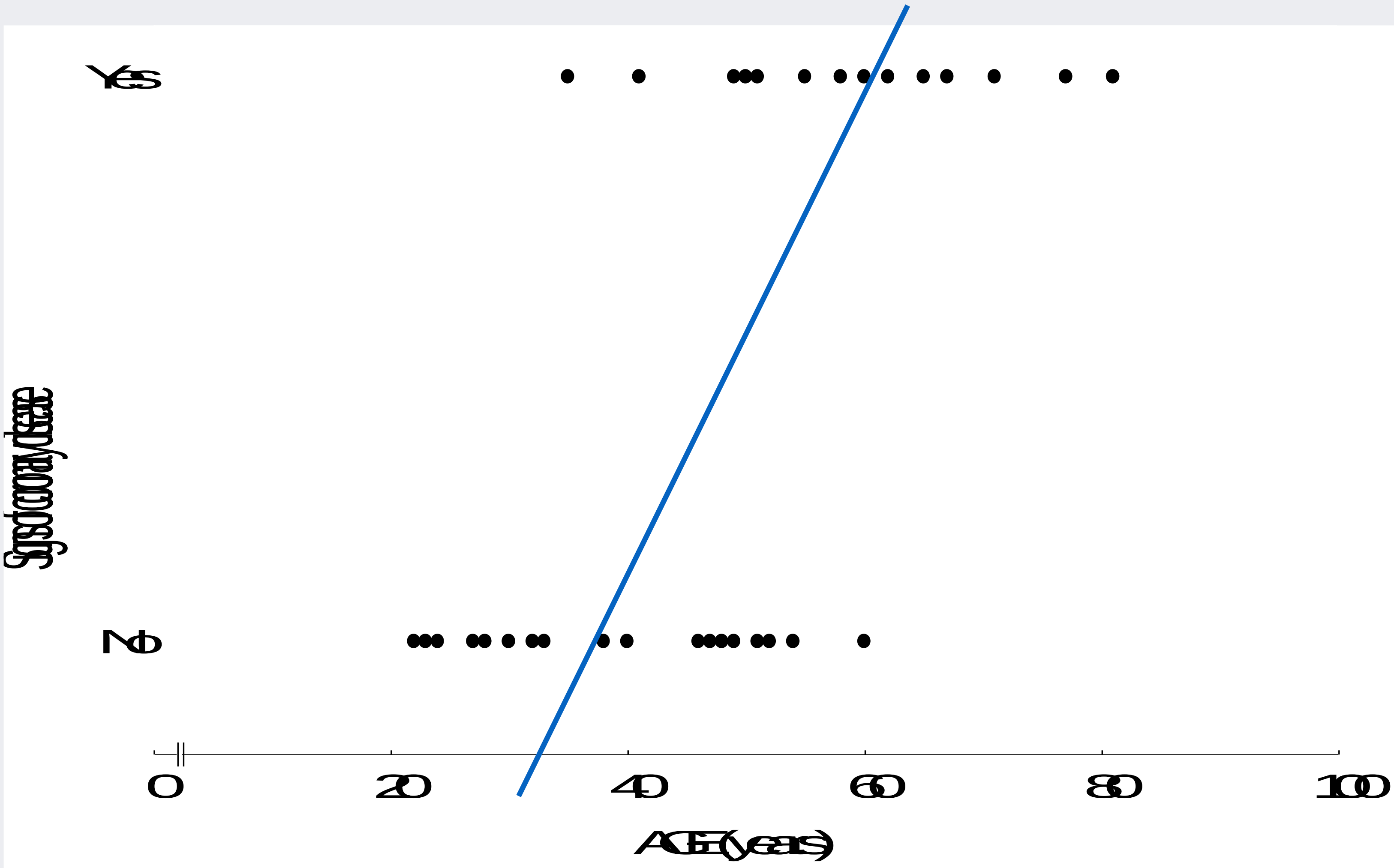
What If

- Example 2
  - ✓ Age and signs of coronary heart disease (CD)

Age	CD	Age	CD	Age	CD
22	0	40	0	54	0
23	0	41	1	55	1
24	0	46	0	58	1
27	0	47	0	60	1
28	0	48	0	60	0
30	0	49	1	62	1
30	0	49	0	65	1
32	0	50	1	67	1
33	0	51	0	71	1
35	1	51	1	77	1
38	0	52	0	81	1

What If

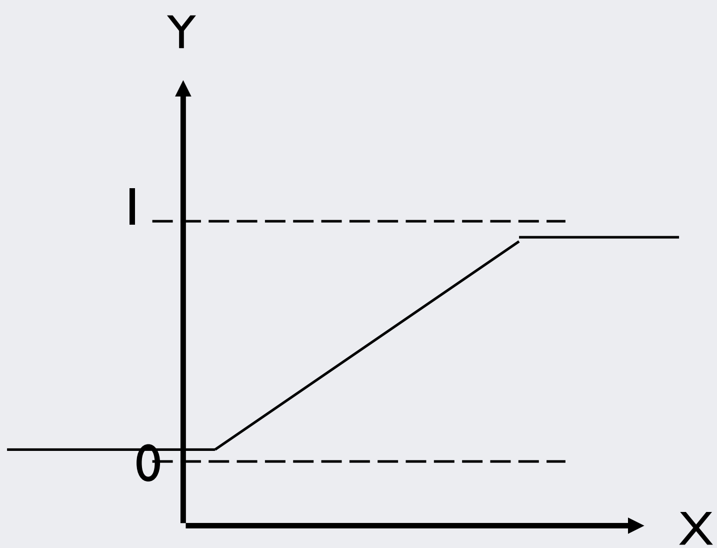
- Linear regression does not estimate Pr(Y=1|X) well



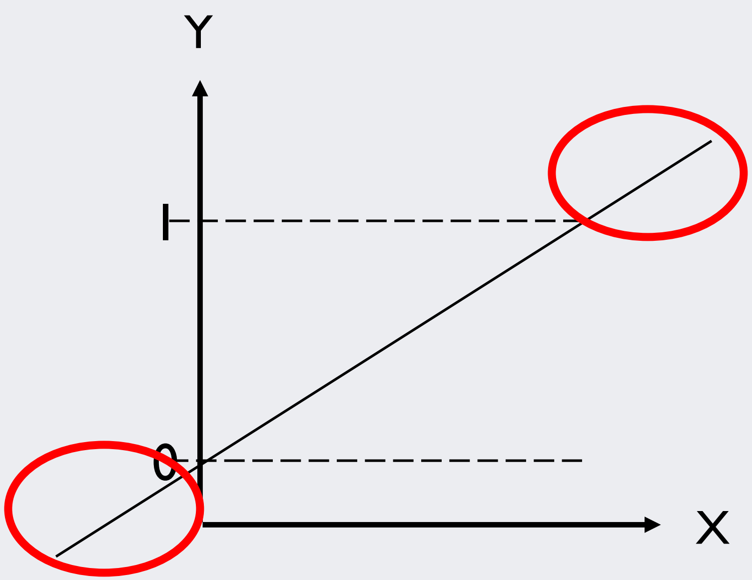
For Classification Task

- Consider when there are only two outcomes (0 & 1)
  - ✓ Is a linear model appropriate?

Ideally:

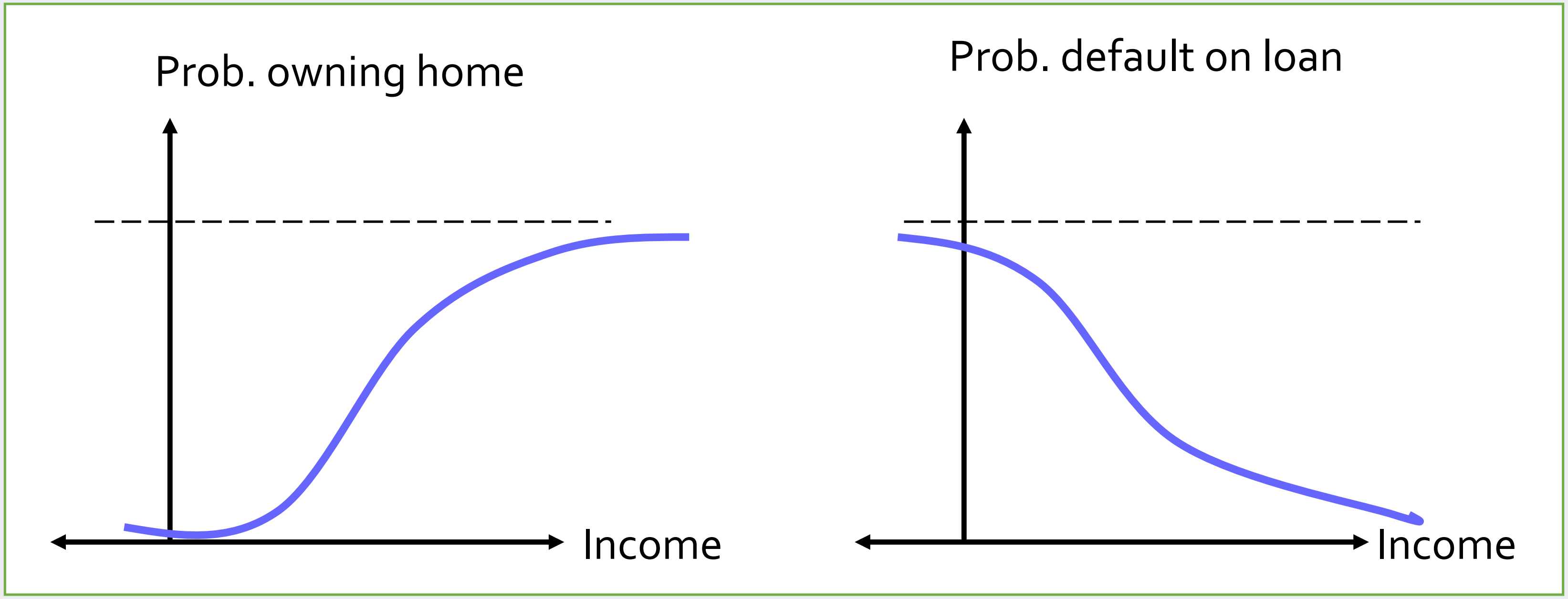


Reality:



For Classification Task

- In real cases…
  - ✓ The probability may follow a certain type of curve rather than a straight line.

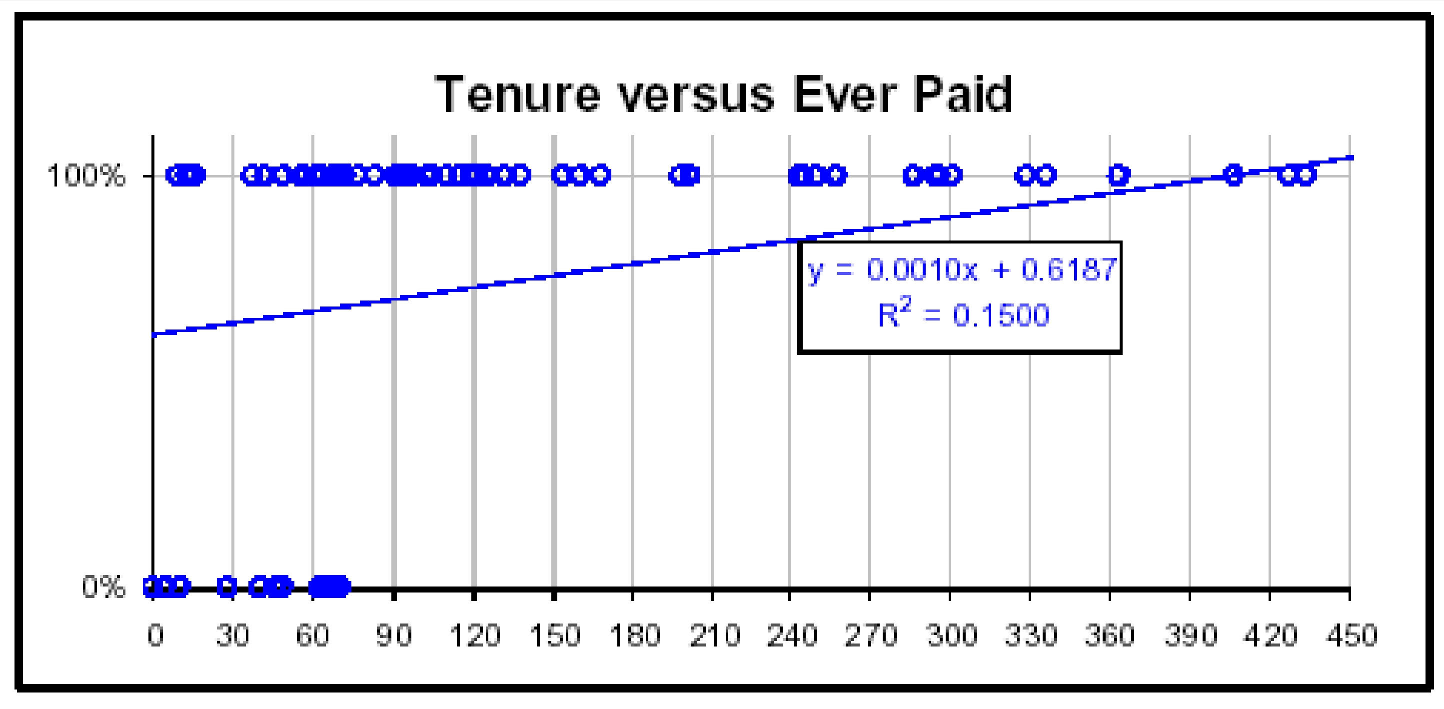


For Classification Task

- Is it appropriate to model the probability as a function of predictors?

$$P(Y=1)=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_p x_p+\varepsilon$$

✓ May have a probability that is greater than 1 or less than 0



Logistic Regression

- Goal:
  - ✓ Find a function of the predictor variables that relates them to a 0/1 outcome

- Features:
  - ✓ Instead of Y as outcome variable (like in linear regression), we use a function of Y called the "logit".
  - ✓ Logit can be modeled as a linear function of the predictors.
  - ✓ The logit can be mapped back to a probability, which, in turn, can be mapped to a class.

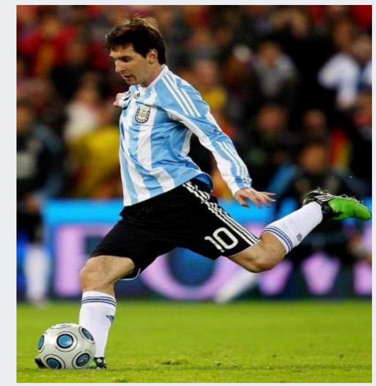
Logistic Regression: Odds

- 2010 World Cup Betting Odds

9 : 2

9 : 2

6 : 1



9 : 1

200 : 1

250 : 1

500 : 1

1000 : 1

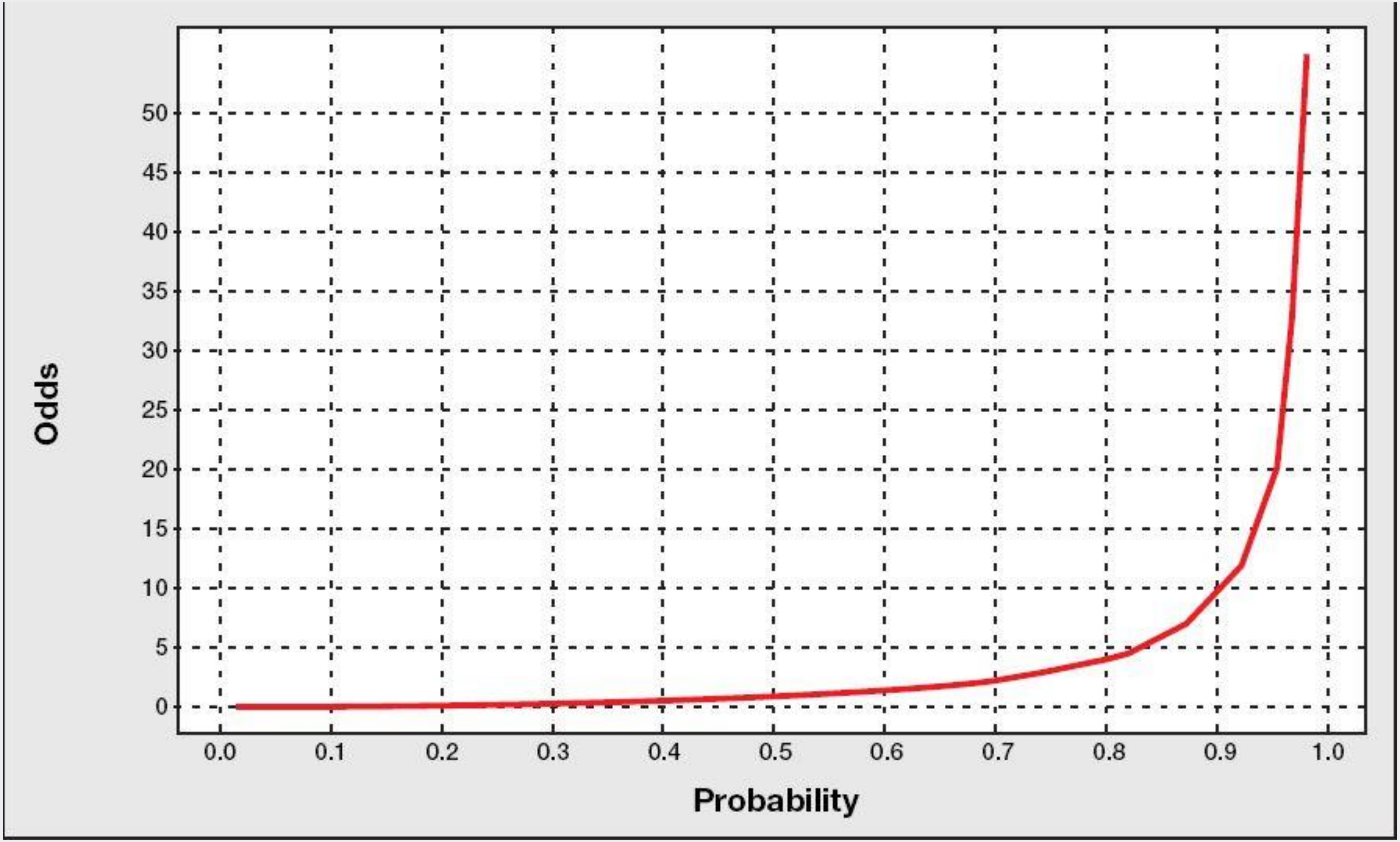
Logistic Regression: Odds

- Odds
  - ✓ p = probability of belonging to class 1 (success).

$$Odds = \frac{p}{1-p}$$

- For the previous examples
  - ✓ Winning odds of the Spain = 2/9, then the winning probability of the Spain = 2/11.
  - ✓ Winning odds of the Korea = 1/250, then the winning probability of the Korea = 1/251 ≒ 0.00398 (0.398%)

Logistic Regression: Odds



Logistic Regression: Log odds

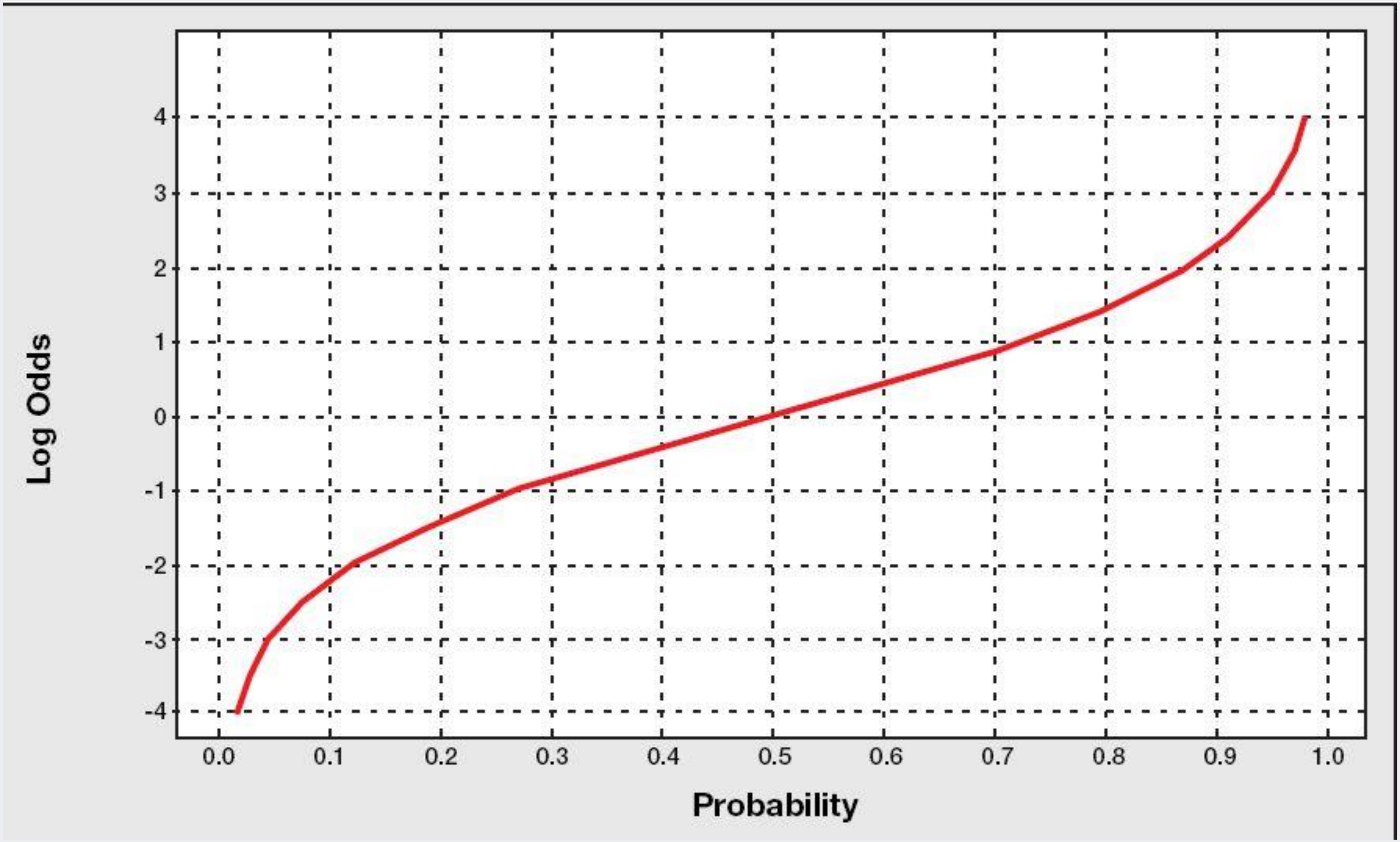
- The limitation of the odds
  - ✓ $0 < \text{odds} < \infty$
  - ✓ Asymmetric

- Take the logarithm of the odds

$$\log(Odds) = \log\left(\frac{p}{1-p}\right)$$

  - ✓ $-\infty < \log(\text{odds}) < \infty$
  - ✓ Symmetric
  - ✓ Negative when p is small and positive when p is large

Logistic Regression: Log odds



Logistic Regression: Equation

- Logistic regression equation
  - ✓ Linear equation for the odds:

$$\log(Odds) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

  - ✓ Take the exponential for the both sides:

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n}$$

  - ✓ For the probability of the success:

$$p = \sigma(\mathbf{x}, \beta) = \frac{1}{1 + e^{-\left(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n\right)}}$$

Logistic Regression: Learning

- Maximum likelihood estimation (MLE)
  - ✓ Expectation function:

$$P(x_i, y_i \mid \beta) = \begin{cases} \sigma(\mathrm{x},\beta) & \text{if } y = 1 \\ 1-\sigma(\mathrm{x},\beta) & \text{if } y = 0 \end{cases}$$

$$= \sigma(\mathrm{x},\beta)^{y}\left(1-\sigma(\mathrm{x},\beta)\right)^{1-y}$$

  - ✓ Likelihood and log-likelihood of the training data **X**:

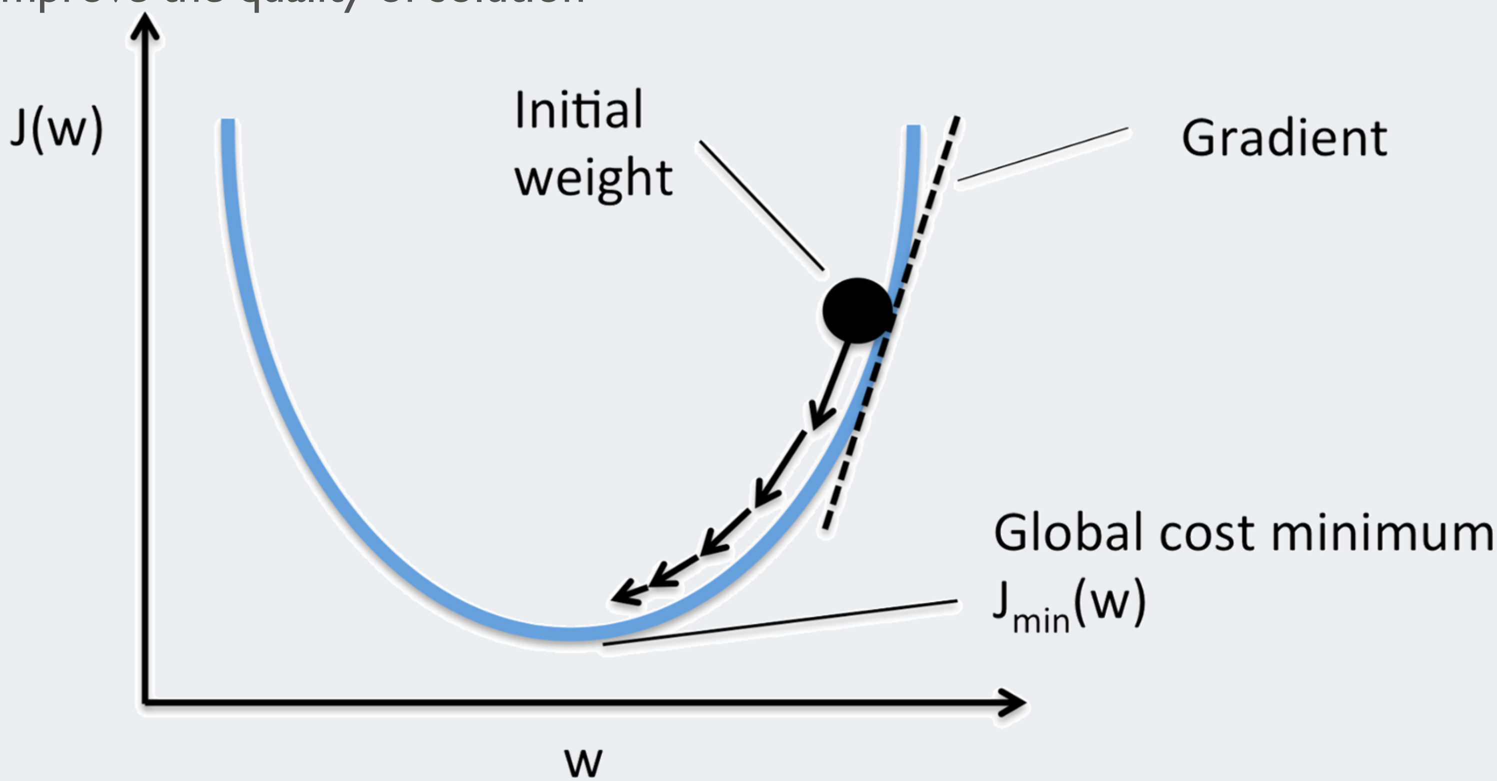
$$L(\mathrm{X}, y, \beta) = \prod_{i=1}^{R} \sigma(\mathrm{x_i},\beta)^{y_i}\left(1-\sigma(\mathrm{x_i},\beta)\right)^{1-y_i}$$

$$\ln L(\mathrm{X}, y, \beta) = \sum_{i=1}^{R} y_i \ln\left(\sigma(\mathrm{x_i},\beta)\right) + (1-y_i)\ln\left(1-\sigma(\mathrm{x_i},\beta)\right)$$

  - ✓ The likelihood and log-likelihood functions are nonlinear in beta and cannot be solved analytically.
    - Numerical methods are typically used to find the MLE.
    - Conjugate gradient is a popular choice.

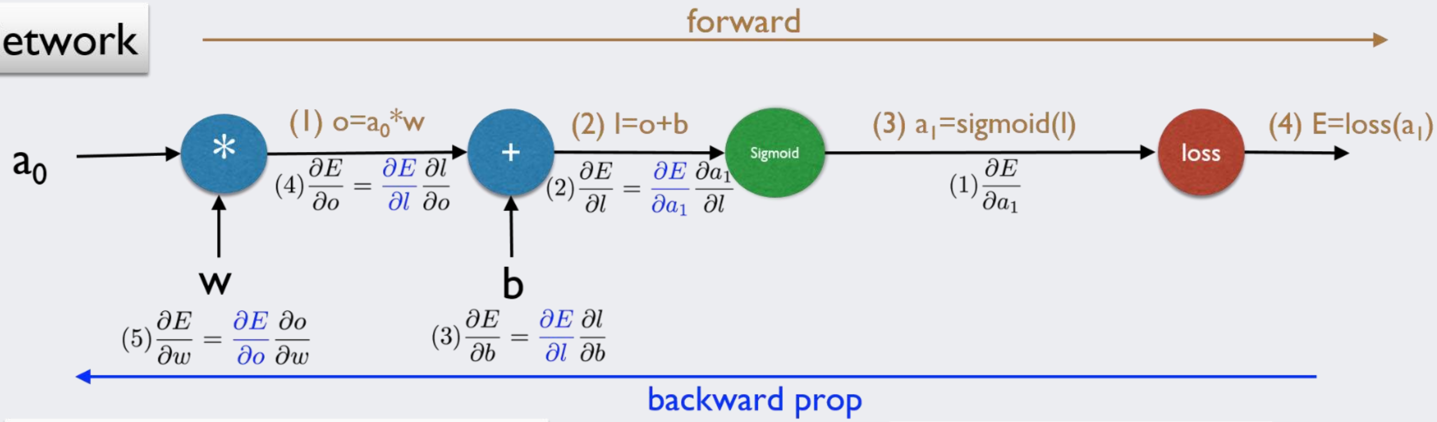
Logistic Regression: Learning

- Gradient Descent Algorithm
  - ✓ Blue line: the objective function to be minimized
  - ✓ Black circle: the current solution
  - ✓ Direction of the arrows: the direction that the current solution should move to improve the quality of solution



Logistic Regression: Learning

Network



Derivatives (chain rule)

$$(1)\frac{\partial E}{\partial a_1} = \frac{a_1 - t}{a_1(1-a_1)}$$

Given from the pre computed derivative

$$(2)\frac{\partial E}{\partial l} = \frac{\partial E}{\partial a_1}\frac{\partial a_1}{\partial l} = \frac{a_1 - t}{a_1(1-a_1)} * a_1(1-a_1) = a_1 - t$$

Given

$$(3)\frac{\partial E}{\partial b} = \frac{\partial E}{\partial l}\frac{\partial l}{\partial b} = \frac{\partial E}{\partial l} * 1 = a_1 - t$$

$$(4)\frac{\partial E}{\partial o} = \frac{\partial E}{\partial l}\frac{\partial l}{\partial o} = \frac{\partial E}{\partial l} * 1 = a_1 - t$$

$$(5)\frac{\partial E}{\partial w} = \frac{\partial E}{\partial o}\frac{\partial o}{\partial w} = \frac{\partial E}{\partial o} a_0 = (a_1 - t)a_0$$

Gate derivatives

$$o = aw,\ \frac{\partial o}{\partial w} = a,\ \frac{\partial o}{\partial a} = w$$

$$l = o + b,\ \frac{\partial l}{\partial o} = 1,\ \frac{\partial l}{\partial b} = 1$$

$$E = -\sum t\,log(a) + (1-t)log(1-a),\ \frac{\partial E}{\partial a} = \frac{a-t}{a(1-a)}$$

$$a = sigmoid(l) = \frac{1}{1+e^{-l}},\ \frac{\partial a}{\partial l} = a(1-a)$$

Network update (learning rate, alpha)

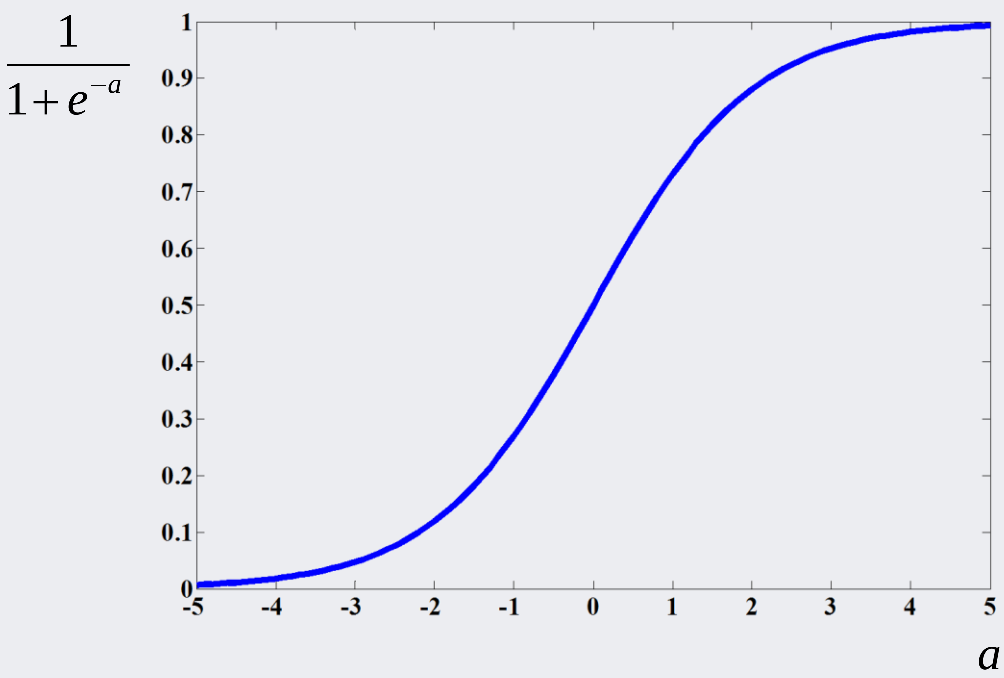
$$w = w - \alpha\frac{\partial E}{\partial w}$$

$$b = b - \alpha\frac{\partial E}{\partial b}$$

Logistic Regression: Prediction

- Success probability
  - ✓ When a set of predictors (independent variables) are given, we can estimate the probability of the success.

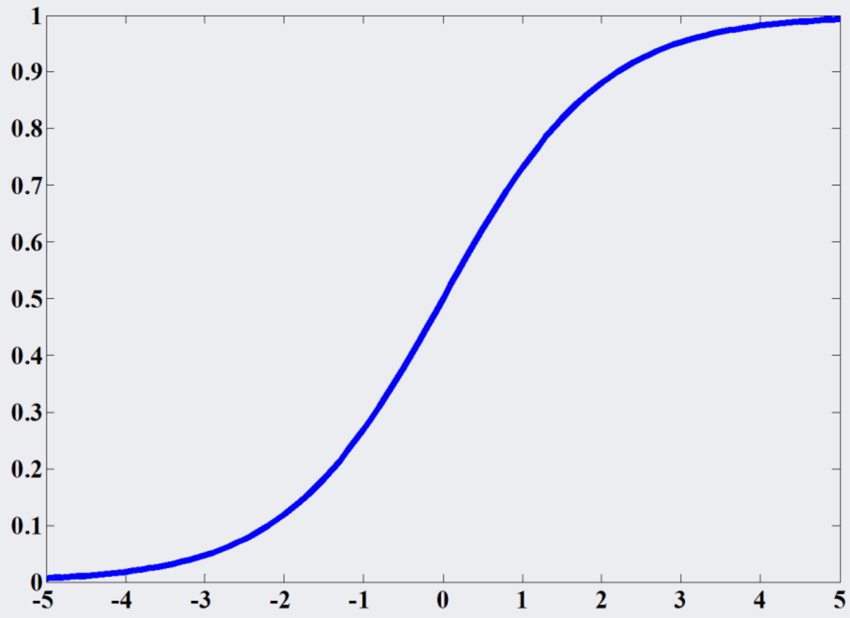
$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}}$$



Logistic Regression: Cut-off

- Determine the cut-off for the binary classification

$$\frac{1}{1+e^{-a}}$$

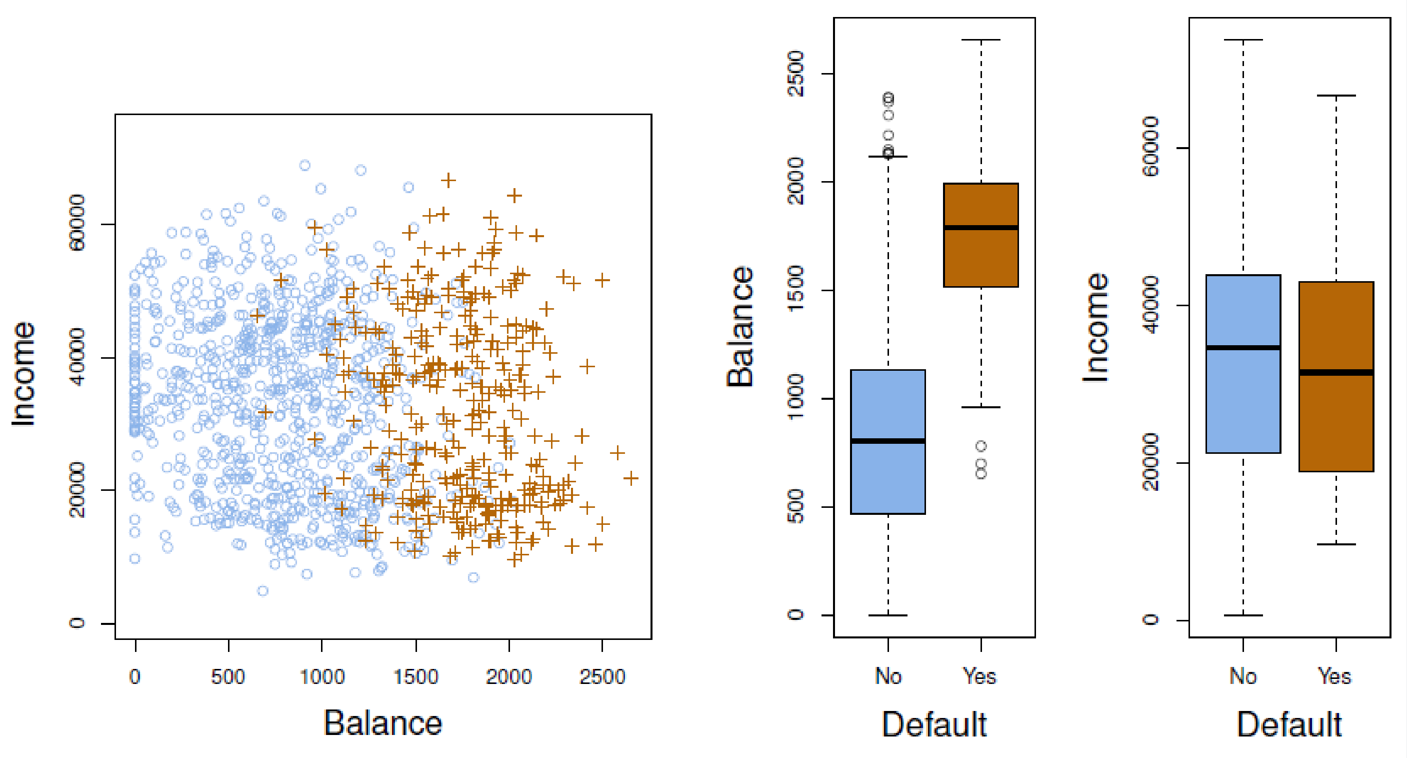


✓ 0.50 is popular initial choice

✓ Additional considerations: max. classification accuracy, max. sensitivity (subject to min. level of specificity), min. false positives (subject to max. false negative rate), min. expected cost of misclassification (need to specify costs)

Logistic Regression: Example 1

- Credit Card Default



$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

Logistic Regression: Example 1

- Credit Card Default: single variable

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

What is our estimated probability of default for someone with a balance of $1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of $2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Logistic Regression: Example 1

- Credit Card Default: multiple variables

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.8690	0.4923	-22.08	< 0.0001
`balance`	0.0057	0.0002	24.74	< 0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

Logistic Regression: Example 2

- Personal Loan Offer
  - ✓ Predict a new customer whether he/she will accept the bank's personal loan offer

일련 번호	나이	경력	소득	가족 수	월별 신용카드 평균사용액	교육 수준	담보부 채권	개인 대출	증권 계좌	CD 계좌	온라인 뱅킹	신용 카드
1	25	1	49	4	1.60	UG	0	No	Yes	No	No	No
2	45	19	34	3	1.50	UG	0	No	Yes	No	No	No
3	39	15	11	1	1.00	UG	0	No	No	No	No	No
4	35	9	100	1	2.70	Grad	0	No	No	No	No	No
5	35	8	45	4	1.00	Grad	0	No	No	No	No	Yes
6	37	13	29	4	0.40	Grad	155	No	No	No	Yes	No
7	53	27	72	2	1.50	Grad	0	No	No	No	Yes	No
8	50	24	22	1	0.30	Prof	0	No	No	No	No	Yes
9	35	10	81	3	0.60	Grad	104	No	No	No	Yes	No
10	34	9	180	1	8.90	Prof	0	Yes	No	No	No	No
11	65	39	105	4	2.40	Prof	0	No	No	No	No	No
12	29	5	45	3	0.10	Grad	0	No	No	No	Yes	No
13	48	23	114	2	3.80	Prof	0	No	Yes	No	No	No
14	59	32	40	4	2.50	Grad	0	No	No	No	Yes	No
15	67	41	112	1	2.00	UG	0	No	Yes	No	No	No
16	60	30	22	1	1.50	Prof	0	No	No	No	Yes	Yes
17	38	14	130	4	4.70	Prof	134	Yes	No	No	No	No
18	42	18	81	4	2.40	UG	0	No	No	No	No	No
19	46	21	193	2	8.10	Prof	0	Yes	No	No	No	No
20	55	28	21	1	0.50	Grad	0	No	Yes	No	No	Yes

Logistic Regression: Example 2

- Data Preprocessing

  - A total of 5,000 customers
  - Predictors
    - ✓ Demographic: age, income, etc.
    - ✓ Relationship with the bank: mortgage, security account, etc.
  - Only 480(9.6%) accepted the personal loan.

  - 60% for training, 40% for validation.
  - Create dummy variables for the categorical predictors.

$$EducProf = \begin{cases} 1 \text{ if education is } \textit{Professional} \\ 0 \text{ otherwise} \end{cases}$$

$$EducGrad = \begin{cases} 1 \text{ if education is at } \textit{Graduate} \text{ level} \\ 0 \text{ otherwise} \end{cases}$$

Logistic Regression: Example 2

- Modeling with all input variables

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}}$$

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

Logistic Regression: Interpretation

- Coefficient
  - ✓ The beta values for corresponding input variables
  - ✓ The value is the changing ratio of log odds when the input variable increases by 1
  - ✓ Positive value: positively correlated with the success class
  - ✓ Negative value: negatively correlated with the success class

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

Logistic Regression: Interpretation

- p-value
  - ✓ Indicating whether the corresponding input variable is statistically significant or not
  - ✓ Significance is strongly supported when the p-value is close to 0

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

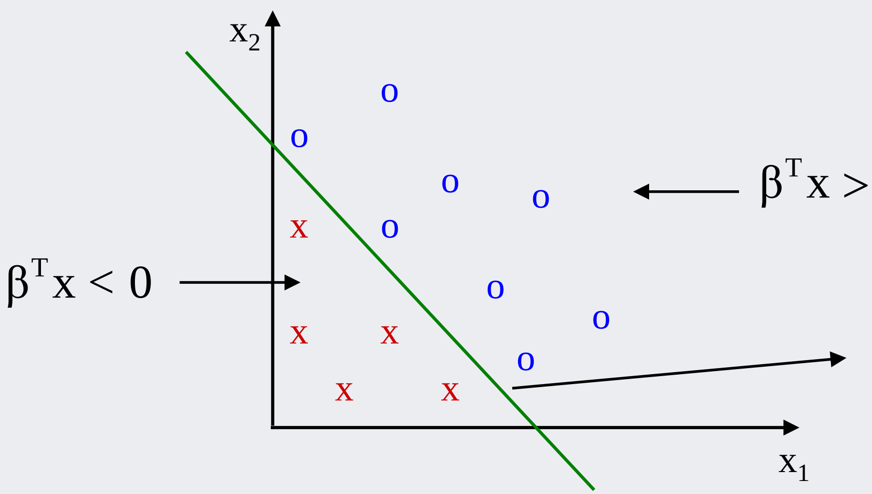
Logistic Regression: Interpretation

- Odds ratio
  - ✓ The ratio of odds when the value of the corresponding input variable increases by 1

Input variables	Coefficient	Std. Error	p-value	Odds
Constant term	-13.20165825	2.46772742	0.00000009	*
Age	-0.04453737	0.09096102	0.62439483	0.95643985
Experience	0.05657264	0.09005365	0.5298661	1.05820346
Income	0.0657607	0.00422134	0	1.06797111
Family	0.57155931	0.10119002	0.00000002	1.77102649
CCAvg	0.18724874	0.06153848	0.00234395	1.20592725
Mortgage	0.00175308	0.00080375	0.02917421	1.00175464
Securities Account	-0.85484785	0.41863668	0.04115349	0.42534789
CD Account	3.46900773	0.44893095	0	32.10486984
Online	-0.84355801	0.22832377	0.00022026	0.43017724
CreditCard	-0.96406376	0.28254223	0.00064463	0.38134006
EducGrad	4.58909273	0.38708162	0	98.40509796
EducProf	4.52272701	0.38425466	0	92.08635712

Logistic Regression: Interpretation

- Geometric interpretation
  - ✓ Can be thought of as finding a hyper-plane to separate positive and negative data points.



Hyperplane: $\beta^T x = 0$ where $\beta = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$

Classifier

$$y = \frac{1}{\left(1 + \exp(-\beta^T x)\right)}$$

$$\begin{pmatrix} y \to 1 & \textit{if} & \beta^T x \to \infty \\ y = \frac{1}{2} & \textit{if} & \beta^T x = 0 \\ y \to 0 & \textit{if} & \beta^T x \to -\infty \end{pmatrix}$$

Logistic Regression: Interpretation

- Odds ratio
  - ✓ Suppose that the value of x_1 is increased by one unit from x_1 to x_1+1, while the other predictors are held at their current value.
  - ✓ Odds ratio:

$$\frac{odds(x_1+1,\ldots,x_n)}{odds(x_1,\ldots,x_n)} = \frac{e^{\beta_0+\beta_1(x_1+1)+\beta_2 x_2+\ldots+\beta_n x_n}}{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n}} = e^{\beta_1}$$

  - ✓ When x_1 is increased by 1, then the odds is increased(decreased) by a factor of e^{β_1}

Logistic Regression: Interpretation

- Profiling
  - ✓ Finding factors that differentiate between the two classes.
  - ✓ After variable selection:

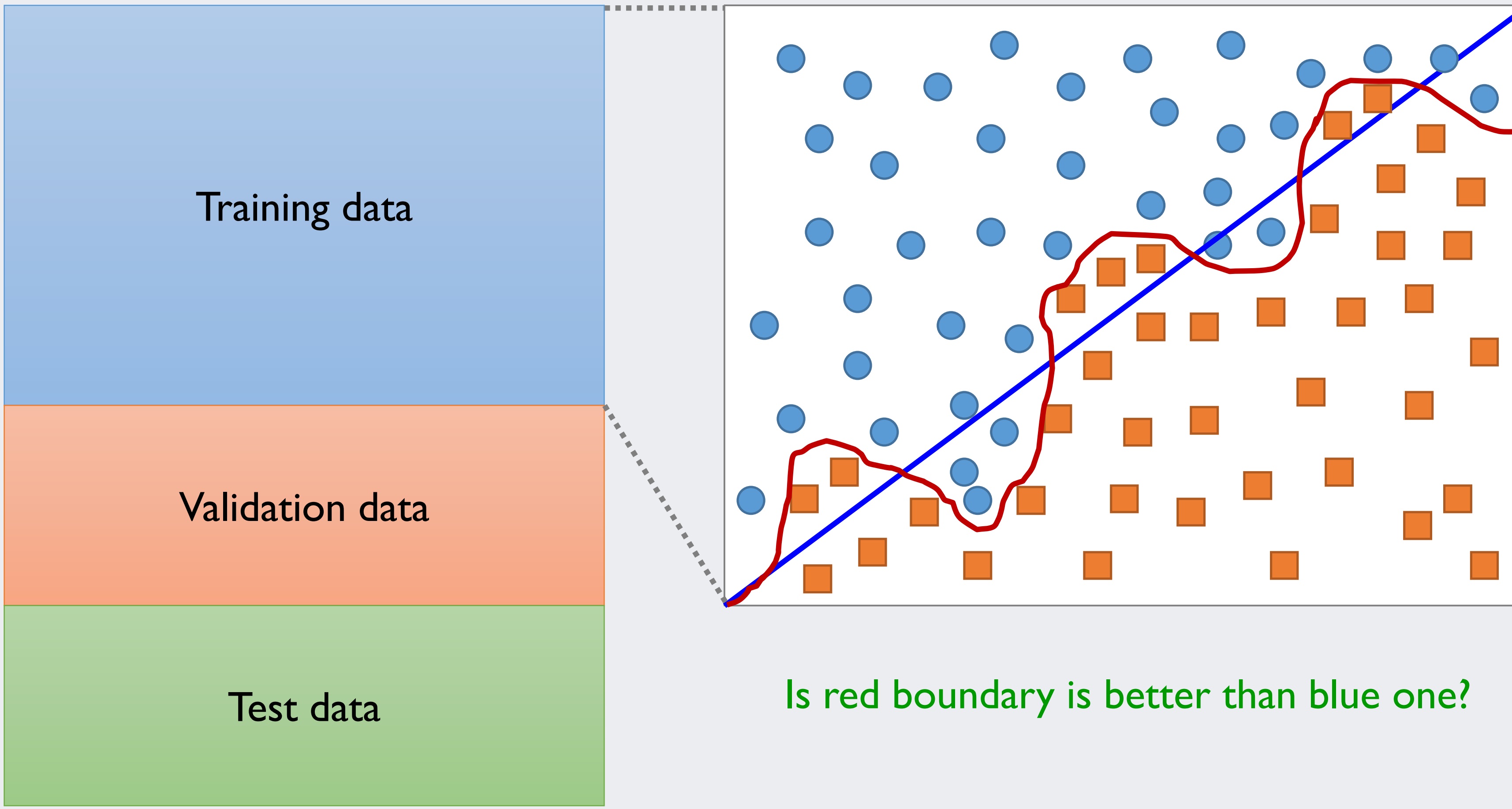
$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d}$$

  - ✓ Variables associated with positive β_i increase the probability of the success.
  - ✓ Variables associated with negative β_i decrease the probability of the success.

AGENDA

Why Evaluate?

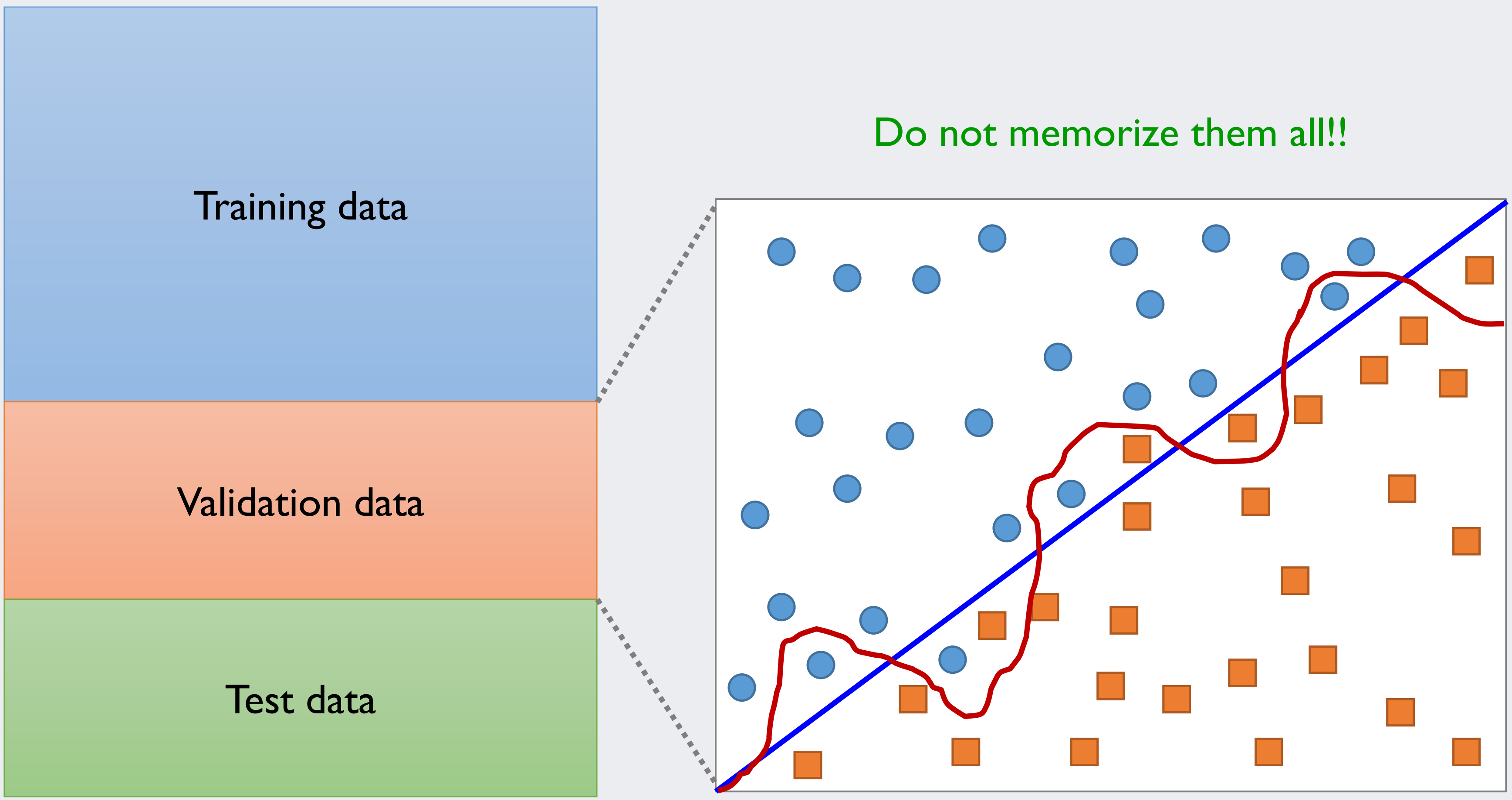
- Over-fitting for training data



Is red boundary is better than blue one?

Why Evaluate?

- Over-fitting for training data



Why Evaluate?

- Multiple methods are available to classify or predict.
  - ✓ Classification:
    - Naïve bayes, linear discriminant, k-nearest neighbor, classification trees, etc.
  - ✓ Prediction:
    - Multiple linear regression, neural networks, regression trees, etc.
- For each method, multiple choices are available for settings.
  - ✓ Neural networks: # hidden nodes, activation functions, etc.
- To choose best model, need to assess each model's performance.
  - ✓ Best setting (parameters) among various candidates for an algorithm (validation).
  - ✓ Best model among various data mining algorithms for the task (test).

Classification Performance

1

Example: Gender classification

- Classify a person based on his/her body fat percentage (BFP).

10.0	21.7	8.9	19.9	23.4	28.9	15.7	21.6	21.5	23.2

- Simple classifier: if BFP > 20 then female else male.

10.0	21.7	8.9	19.9	23.4	28.9	15.7	21.6	21.8	23.2
M	F	M	M	F	F	M	F	F	F

- How do you evaluate the performance of the above classifier?

Classification Performance

2

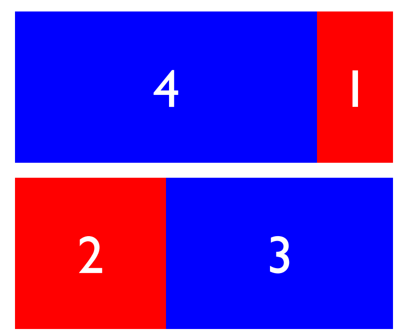
Confusion Matrix

- Summarizes the correct and incorrect classifications that a classifier produced for a certain data set.

10.0	21.7	8.9	19.9	23.4	28.9	15.7	21.6	21.5	23.2
M	F	M	M	F	F	M	F	F	F

- Confusion matrix can be constructed as

Confusion Matrix		Predicted F	Predicted M
Actual	F	4	1
Actual	M	2	3



Classification Performance

2

Confusion Matrix

- Summarizes the correct and incorrect classifications that a classifier produced for a certain data set.

Confusion Matrix		Predicted	
		1(+)	0(-)
Actual	1(+)	n_{11}	n_{10}
	0(-)	n_{01}	n_{00}

Confusion Matrix		Predicted	
		F	M
Actual	F	4	1
	M	2	3

- Misclassification error = $(n_{01}+n_{10})/(n_{11}+n_{10}+n_{01}+n_{00})$ = (2+1)/10 = 0.3
- Accuracy = (1-Misclassification error) = $(n_{11}+n_{00})/(n_{11}+n_{10}+n_{01}+n_{00})$ = (4+3)/10 = 0.7

Classification Performance

2

Confusion Matrix

- Summarizes the correct and incorrect classifications that a classifier produced for a certain data set.

Confusion Matrix		Predicted	
		1(+)	0(-)
Actual	1(+)	n_{11}	n_{10}
	0(-)	n_{01}	n_{00}

Confusion Matrix		Predicted	
		F	M
Actual	F	4	1
	M	2	3

- Balanced correction rate (BCR): $\sqrt{\frac{n_{11}}{n_{11}+n_{10}} \cdot \frac{n_{00}}{n_{01}+n_{00}}}$ $= \sqrt{0.8 \times 0.6}$ = 0.69

- F1-Measure: $\frac{2 \times Recall \times Precision}{Recall + Precision} = \frac{2 \times 0.8 \times 0.67}{0.8 + 0.67} = 0.85$

Classification Performance

3

Cut-off for classification

- A new classifier: : if BFP > θ then female else male.

10.0	21.7	8.9	19.9	23.4	28.9	15.7	21.6	21.5	23.2

- Sort data in a descending order of BFS.

28.6	25.4	24.2	23.6	22.7	21.5	19.9	15.7	10.0	8.9

- How do you decide the cut-off for classification?

Classification Performance

3

Cut-off for classification

- Performance measures for different cut-offs:

No.	BFS	Gender
1	28.6	F
2	25.4	M
3	24.2	F
4	23.6	F
5	22.7	F
6	21.5	M
7	19.9	F
8	15.7	M
9	10.0	M
10	8.9	M

- If $\theta = 24$,

Confusion Matrix		Predicted F	Predicted M
Actual	F	2	3
	M	1	4

- Misclassification error: 0.4
- Accuracy: 0.6
- Balanced correction rate: 0.57
- F1 measure = 0.5

Classification Performance

3

Cut-off for classification

- Performance measures for different cut-offs:

No.	BFS	Gender
1	28.6	F
2	25.4	M
3	24.2	F
4	23.6	F
5	22.7	F
6	21.5	M
7	19.9	F
8	15.7	M
9	10.0	M
10	8.9	M

- If θ = 22,

Confusion Matrix		Predicted	
		F	M
Actual	F	4	1
	M	1	4

- Misclassification error: 0.2
- Accuracy: 0.8
- Balanced correction rate: 0.8
- F1 measure = 0.8

Classification Performance

3

Cut-off for classification

- Performance measures for different cut-offs:

No.	BFS	Gender
1	28.6	F
2	25.4	M
3	24.2	F
4	23.6	F
5	22.7	F
6	21.5	M
7	19.9	F
8	15.7	M
9	10.0	M
10	8.9	M

- If $\theta = 18$,

Confusion Matrix		Predicted	
		F	M
Actual	F	5	0
	M	2	3

- Misclassification error: 0.2
- Accuracy: 0.8
- Balanced correction rate: 0.77
- F1 measure = 0.83

Classification Performance

3

Cut-off for classification

- In general, classification algorithms can produce the likelihood for each class in terms of probability or degree of evidence, etc.
- Classification performance highly depends on the cut-off of the algorithm.
- For model selection & model comparison, cut-off independent performance measures are recommended.
- Lift charts, receiver operating characteristic (ROC) curve, etc.

Classification Performance

- Area Under Receiver Operating Characteristic Curve (AUROC)
  - ✓ Cancer diagnosis:
    - Predict patients' probability of malignant.
    - A total of 100 patients.
    - 20 patients are malignant.
    - Malignant ratio: 0.2.

Patient	P(Malignant)	Status	Patient	P(Malignant)	Status	Patient	P(Malignant)	Status	Patient	P(Malignant)	Status
1	0.976	1	26	0.716	1	51	0.410	0	76	0.186	0
2	0.973	1	27	0.676	0	52	0.406	1	77	0.183	0
3	0.971	0	28	0.672	0	53	0.378	0	78	0.178	0
4	0.967	1	29	0.662	0	54	0.376	0	79	0.178	0
5	0.937	0	30	0.647	0	55	0.362	0	80	0.173	0
6	0.936	1	31	0.640	1	56	0.355	0	81	0.170	0
7	0.929	1	32	0.625	0	57	0.343	0	82	0.133	0
8	0.927	0	33	0.624	0	58	0.338	0	83	0.120	0
9	0.923	1	34	0.613	1	59	0.335	0	84	0.119	0
10	0.898	0	35	0.606	0	60	0.334	0	85	0.112	0
11	0.863	1	36	0.604	0	61	0.328	0	86	0.093	0
12	0.863	1	37	0.601	0	62	0.313	0	87	0.086	0
13	0.859	0	38	0.594	0	63	0.285	1	88	0.079	0
14	0.855	0	39	0.578	0	64	0.274	0	89	0.071	0
15	0.847	1	40	0.548	0	65	0.274	0	90	0.069	0
16	0.847	1	41	0.539	1	66	0.272	0	91	0.047	0
17	0.837	0	42	0.525	1	67	0.267	0	92	0.029	0
18	0.833	0	43	0.524	0	68	0.265	0	93	0.028	0
19	0.814	0	44	0.514	0	69	0.237	0	94	0.027	0
20	0.813	0	45	0.510	0	70	0.217	0	95	0.022	0
21	0.793	1	46	0.509	0	71	0.213	0	96	0.019	0
22	0.787	0	47	0.455	0	72	0.204	1	97	0.015	0
23	0.757	1	48	0.449	0	73	0.201	0	98	0.010	0
24	0.741	0	49	0.434	0	74	0.200	0	99	0.005	0
25	0.737	0	50	0.414	0	75	0.193	0	100	0.002	0

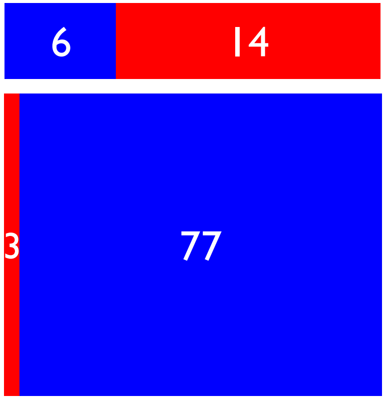
Classification Performance

4

Confusion matrix

- Set the cut-off to 0.9
  - Malignant if P(Malignant) > 0.9, else benign.

Confusion Matrix		Predicted M	Predicted B
Actual	M	6	14
	B	3	77



  - Misclassification error = 0.17
  - Accuracy = 0.83
- Is it a good classification model?

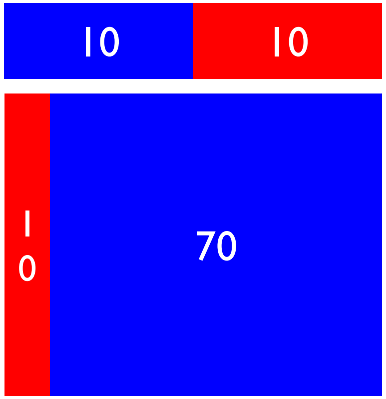
Classification Performance

4

Confusion matrix

- Set the cut-off to 0.8
  - Malignant if P(Malignant) > 0.8, else benign.

Confusion Matrix		Predicted M	Predicted B
Actual	M	10	10
	B	10	70



  - Misclassification error = 0.2
  - Accuracy = 0.8
- Is it worse than the previous model?

Classification Performance

Receiver operating characteristics (ROC) curve

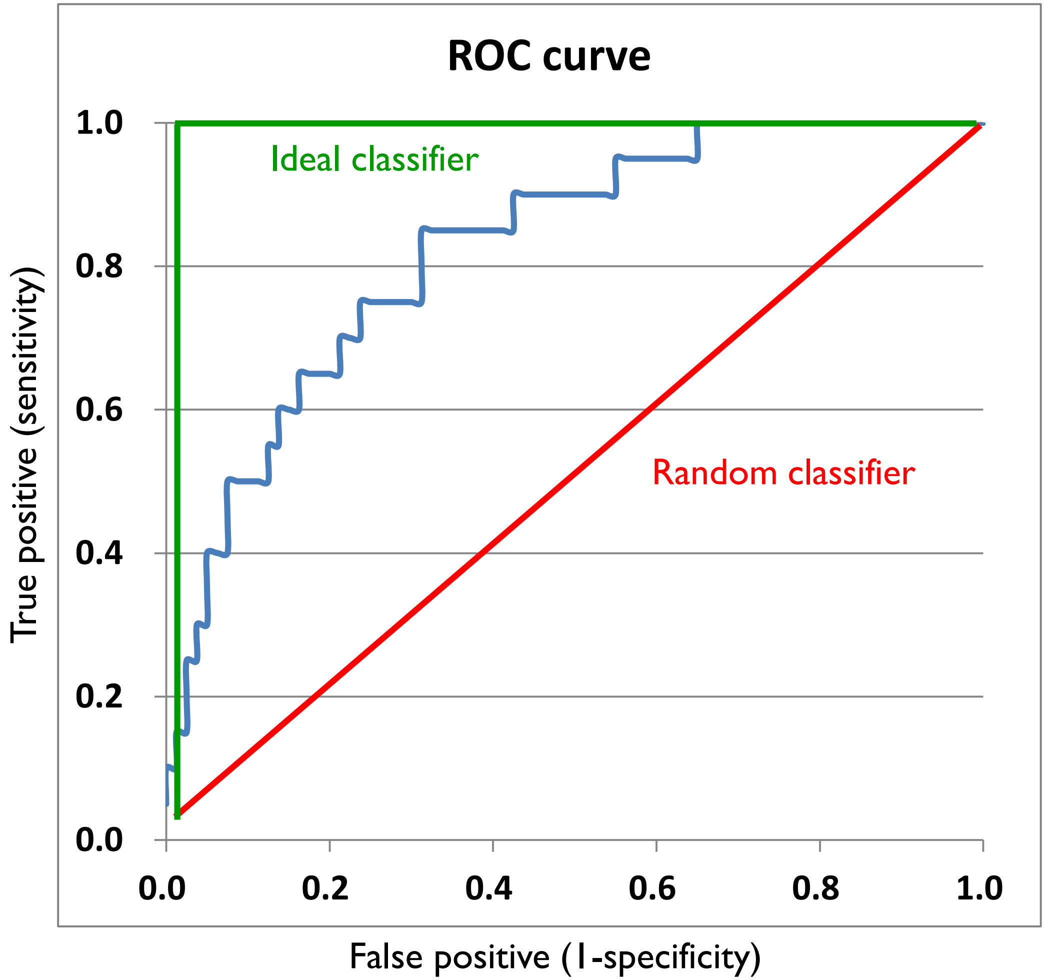
- Sort the records based on the P(interesting class) in a descending order.
- Compute the true positive rate and false positive rate by varying the cut-off.
- Draw a chart where x & y axes are false & true positive, respectively.

Patient	P(Malignant)	Status	True positive	false positive
1	0.976	1	0.050	0.000
2	0.973	1	0.100	0.000
3	0.971	0	0.100	0.013
4	0.967	1	0.150	0.013
5	0.937	0	0.150	0.025
6	0.936	1	0.200	0.025
7	0.929	1	0.250	0.025
8	0.927	0	0.250	0.038
⋮	⋮	⋮	⋮	⋮
96	0.019	0	1.000	0.950
97	0.015	0	1.000	0.963
98	0.010	0	1.000	0.975
99	0.005	0	1.000	0.988
100	0.002	0	1.000	1.000

5

Classification Performance

Receiver operating characteristics (ROC) curve

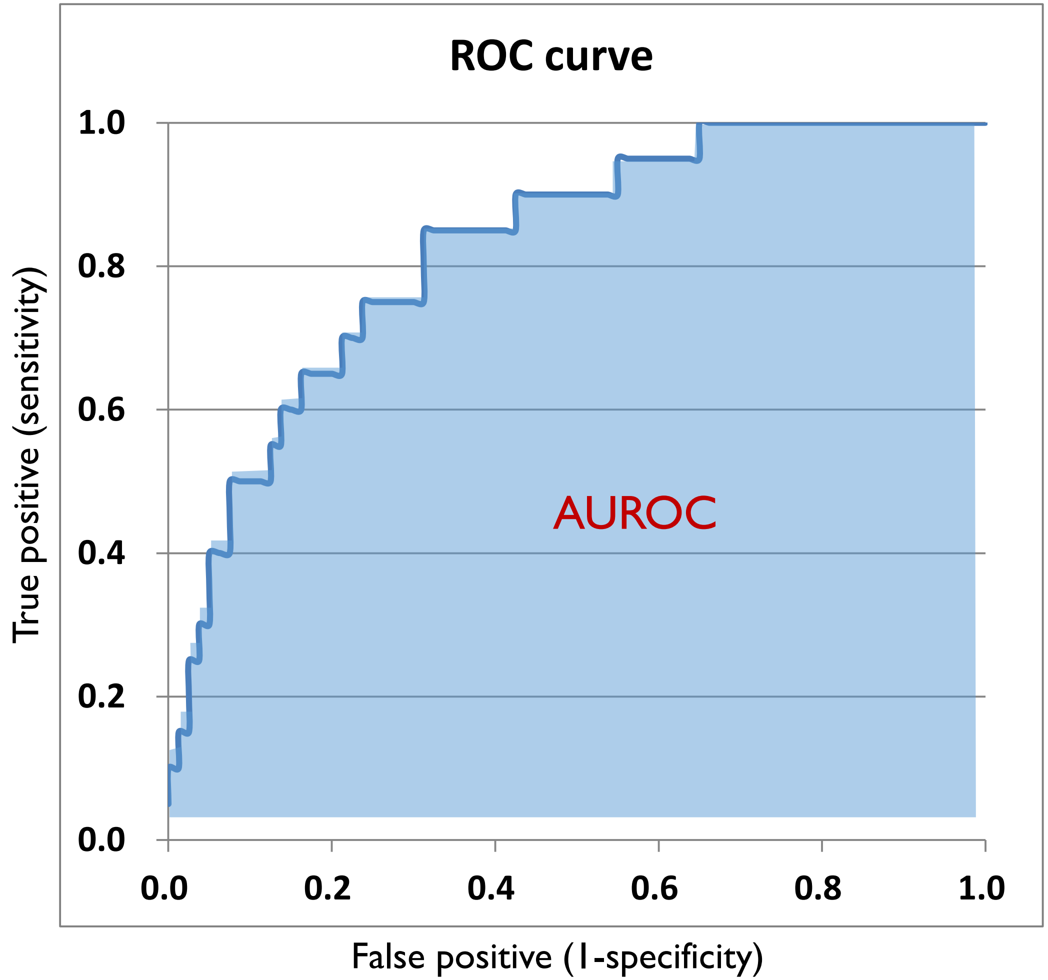


5

Classification Performance

Area Under ROC curve (AUROC)

- The area under the ROC curve.
- Can be a useful metric for parameter/model selection.
- 1 for the ideal classifier
- 0.5 for the random classifier.



6

AGENDA

R Exercise

- Data Set: Personal Loan Prediction

Data Description:

ID	Customer ID
Age	Customer's Age in completed years
Experience	#years of professional experience
Income	Annual income of the customer ($000)
ZIPCode	Home Address ZIP code.
Family	Family size (dependents) of the customer
CCAvg	Avg. Spending on Credit Cards per month ($000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. ($000)
Personal Loan	Did this customer accept the personal loan offered in the last campaign?
Securities Account	Does the customer have a Securities account with the bank?
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by UniversalBank?

R Exercise

- Create a performance evaluation function
  - ✓ True positive rate, Precision, True negative rate, Accuracy, Balance correction rate, and F1-measure

```
# Performance Evaluation Function ----------------------------------------
perf_eval <- function(cm){

  # True positive rate: TPR (Recall)
  TPR <- cm[2,2]/sum(cm[2,])

  # Precision
  PRE <- cm[2,2]/sum(cm[,2])

  # True negative rate: TNR
  TNR <- cm[1,1]/sum(cm[1,])

  # Simple Accuracy
  ACC <- (cm[1,1]+cm[2,2])/sum(cm)

  # Balanced Correction Rate
  BCR <- sqrt(TPR*TNR)

  # F1-Measure
  F1 <- 2*TPR*PRE/(TPR+PRE)

  return(c(TPR, PRE, TNR, ACC, BCR, F1))
}
```

R Exercise

- Initialize the performance comparison matrix
  - ✓ Four models and six performance metrics

```
# Evaluate the performance
perf_mat <- matrix(0, 4, 6)
rownames(perf_mat) <- c("LR_ALL", "LR_Forward", "LR_Backward", "LR_Stepwise")
colnames(perf_mat) <- c("TPR", "Precision", "TNR", "ACC", "BCR", "F1")
perf_mat
```

```
> perf_mat
            TPR Precision TNR ACC BCR F1
LR_ALL        0         0   0   0   0  0
LR_Forward    0         0   0   0   0  0
LR_Backward   0         0   0   0   0  0
LR_Stepwise   0         0   0   0   0  0
```

R Exercise

- Load the data and do some preprocessing
  - ✓ Remove unnecessary columns
  - ✓ Normalize the input data

```
# Logistic Regression -----------------------------------------------------
# Conduct the normalization
ploan <- read.csv("Personal Loan.csv")

input_idx <- c(2:9,11:14)
target_idx <- 10

# Select the input variables and normalize them
ploan_input <- ploan[,input_idx]
ploan_input <- scale(ploan_input, center = TRUE, scale = TRUE)
```

R Exercise

- Load the data and do some preprocessing
  - ✓ Remove unnecessary columns
  - ✓ Normalize the input data

	Age	Experience	Income	ZIP.Code	Family	CCAvg	Education	Mortgage	Securities.Account	CD.Account	Online	CreditCard
1	25	1	49	91107	4	1.60	1	0	1	0	0	0
2	45	19	34	90089	3	1.50	1	0	1	0	0	0
3	39	15	11	94720	1	1.00	1	0	0	0	0	0
4	35	9	100	94112	1	2.70	2	0	0	0	0	0
5	35	8	45	91330	4	1.00	2	0	0	0	0	1

	Age	Experience	Income	ZIP.Code	Family	CCAvg	Education	Mortgage	Securities.Account	CD.Account	Online	CreditCard
1	-1.76621927	-1.65924085	-0.54523016	-0.838037972	1.3725800	-0.19565243	-1.0347255	-0.5692367	2.8497084	-0.2579269	-1.2183987	-0.6395944
2	-0.03003597	-0.09725938	-0.86661924	-1.258566482	0.5104066	-0.25134880	-1.0347255	-0.5692367	2.8497084	-0.2579269	-1.2183987	-0.6395944
3	-0.55089096	-0.44436637	-1.35941584	0.654466455	-1.2139401	-0.52983064	-1.0347255	-0.5692367	-0.3507727	-0.2579269	-1.2183987	-0.6395944
4	-0.89812762	-0.96502686	0.54749273	0.403306009	-1.2139401	0.41700762	0.1606598	-0.5692367	-0.3507727	-0.2579269	-1.2183987	-0.6395944
5	-0.89812762	-1.05180361	-0.63093391	-0.745918269	1.3725800	-0.52983064	0.1606598	-0.5692367	-0.3507727	-0.2579269	-1.2183987	1.5628656

R Exercise

- Load the data and do some preprocessing
  - ✓ Remove unnecessary columns
  - ✓ Normalize the input data
  - ✓ Divide the entire dataset into training/validation data sets

Data	
perf_mat	num [1:4, 1:6] 0 0 0 0 0 0 0 0 0 0 ...
ploan	2500 obs. of 14 variables
ploan_data	2500 obs. of 13 variables
ploan_input	num [1:2500, 1:12] -1.766 -0.03 -0.551 -0.898 -0.898 ...
ploan_trn	1750 obs. of 13 variables
ploan_val	750 obs. of 13 variables

R Exercise

- Training the logistic regression with all input variables

```
# Train the Logistic Regression Model with all variables
full_lr <- glm(ploan_target ~ ., family=binomial, ploan_trn)
full_lr
summary(full_lr)
```

R Exercise

- Training the logistic regression with all input variables

```
Call:
glm(formula = ploan_target ~ ., family = binomial, data = ploan_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0690  -0.1923  -0.0758  -0.0288   4.0324

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.79826    0.28968 -16.564  < 2e-16 ***
Age                 0.22769    1.17266   0.194 0.846046
Experience          0.20432    1.16936   0.175 0.861294
Income              2.52408    0.20345  12.406  < 2e-16 ***
ZIP.Code            0.27609    0.17535   1.574 0.115377
Family              0.97499    0.15276   6.383 1.74e-10 ***
CCAvg               0.29092    0.11601   2.508 0.012154 *
Education           1.40976    0.16379   8.607  < 2e-16 ***
Mortgage            0.06476    0.09335   0.694 0.487823
Securities.Account -0.38459    0.16200  -2.374 0.017597 *
CD.Account          0.87333    0.13414   6.510 7.50e-11 ***
Online             -0.18526    0.12966  -1.429 0.153052
CreditCard         -0.53669    0.15737  -3.410 0.000649 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1115.66  on 1749  degrees of freedom
Residual deviance:  444.36  on 1737  degrees of freedom
AIC: 470.36

Number of Fisher Scoring iterations: 8
```

R Exercise

- Training the logistic regression with forward selection

```r
# Train the Logistic Regression Model with forward selection
tmp_x <- paste(colnames(ploan_trn)[-13], collapse=" + ")
tmp_xy <- paste("ploan_target ~ ", tmp_x, collapse = "")
tmp_xy
as.formula(tmp_xy)

forward_lr <- step(lm(ploan_target ~ 1, data = ploan_trn),
                   scope = list(upper = as.formula(tmp_xy), lower = Price ~ 1),
                   direction="forward", trace=1)
summary(forward_lr)
```

R Exercise

- Training the logistic regression with forward selection

```
Call:
lm(formula = ploan_target ~ Income + CD.Account + Education +
    Family + CreditCard + Securities.Account + Experience + CCAvg,
    data = ploan_trn)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66518 -0.12827 -0.03063  0.07503  1.05767

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.098006   0.005539  17.694  < 2e-16 ***
Income              0.144602   0.007359  19.650  < 2e-16 ***
CD.Account          0.079310   0.006363  12.464  < 2e-16 ***
Education           0.065319   0.005623  11.617  < 2e-16 ***
Family              0.044457   0.005669   7.842 7.64e-15 ***
CreditCard         -0.022695   0.005669  -4.003 6.52e-05 ***
Securities.Account -0.019036   0.005828  -3.266  0.00111 **
Experience          0.015075   0.005582   2.701  0.00699 **
CCAvg               0.017571   0.007231   2.430  0.01520 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2314 on 1741 degrees of freedom
Multiple R-squared:  0.3926,    Adjusted R-squared:  0.3898
F-statistic: 140.7 on 8 and 1741 DF,  p-value: < 2.2e-16
```

R Exercise

- Training the logistic regression with backward elimination

```
# Train the Logistic Regression Model with backward elimination
backward_lr <- step(full_lr, scope = list(upper = as.formula(tmp_xy), lower = ploan_target ~ 1),
                    direction="backward", trace=1)
summary(backward_lr)
```

R Exercise

- Training the logistic regression with backward elimination

```
Call:
glm(formula = ploan_target ~ Age + Income + ZIP.Code + Family +
    CCAvg + Education + Securities.Account + CD.Account + Online +
    CreditCard, family = binomial, data = ploan_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1018  -0.1915  -0.0764  -0.0289   4.0563

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -4.7976     0.2893 -16.583  < 2e-16 ***
Age                  0.4332     0.1312   3.302 0.000961 ***
Income               2.5413     0.2021  12.572  < 2e-16 ***
ZIP.Code             0.2723     0.1752   1.555 0.120021
Family               0.9853     0.1525   6.460 1.05e-10 ***
CCAvg                0.2842     0.1156   2.459 0.013933 *
Education            1.3952     0.1604   8.700  < 2e-16 ***
Securities.Account  -0.3854     0.1622  -2.376 0.017485 *
CD.Account           0.8770     0.1342   6.533 6.45e-11 ***
Online              -0.1847     0.1295  -1.426 0.153854
CreditCard          -0.5406     0.1572  -3.440 0.000582 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1115.66  on 1749  degrees of freedom
Residual deviance:  444.86  on 1739  degrees of freedom
AIC: 466.86

Number of Fisher Scoring iterations: 8
```

R Exercise

- Training the logistic regression with stepwise selection

```
# Train the Logistic Regression Model with stepwise selection
stepwise_lr <- step(lm(ploan_target ~ 1, data = ploan_trn),
                   scope = list(upper = as.formula(tmp_xy), lower = Price ~ 1),
                   direction="both", trace=1)
summary(stepwise_lr)
```

R Exercise

- Training the logistic regression with stepwise selection

```
Call:
lm(formula = ploan_target ~ Income + CD.Account + Education +
    Family + CreditCard + Securities.Account + Experience + CCAvg,
    data = ploan_trn)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66518 -0.12827 -0.03063  0.07503  1.05767

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.098006   0.005539  17.694  < 2e-16 ***
Income              0.144602   0.007359  19.650  < 2e-16 ***
CD.Account          0.079310   0.006363  12.464  < 2e-16 ***
Education           0.065319   0.005623  11.617  < 2e-16 ***
Family              0.044457   0.005669   7.842 7.64e-15 ***
CreditCard         -0.022695   0.005669  -4.003 6.52e-05 ***
Securities.Account -0.019036   0.005828  -3.266  0.00111 **
Experience          0.015075   0.005582   2.701  0.00699 **
CCAvg               0.017571   0.007231   2.430  0.01520 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2314 on 1741 degrees of freedom
Multiple R-squared:  0.3926,    Adjusted R-squared:  0.3898
F-statistic: 140.7 on 8 and 1741 DF,  p-value: < 2.2e-16
```

R Exercise

- Evaluate the classification performances

```r
# Evaluate the logistic regression performance on the validation data
# Case 1: full model
full_response <- predict(full_lr, type = "response", newdata = ploan_val)
full_target <- ploan_val$ploan_target
full_predicted <- rep(0, length(full_target))
full_predicted[which(full_response >= 0.5)] <- 1
cm_full <- table(full_target, full_predicted)
perf_mat[1,] <- perf_eval(cm_full)

# Case 2: forward model
forward_response <- predict(forward_lr, type = "response", newdata = ploan_val)
forward_target <- ploan_val$ploan_target
forward_predicted <- rep(0, length(forward_target))
forward_predicted[which(forward_response >= 0.5)] <- 1
cm_forward <- table(forward_target, forward_predicted)
perf_mat[2,] <- perf_eval(cm_forward)

# Case 3: backward
backward_response <- predict(backward_lr, type = "response", newdata = ploan_val)
backward_target <- ploan_val$ploan_target
backward_predicted <- rep(0, length(backward_target))
backward_predicted[which(backward_response >= 0.5)] <- 1
cm_backward <- table(backward_target, backward_predicted)
perf_mat[3,] <- perf_eval(cm_backward)

# Case 4: stepwise
stepwise_response <- predict(stepwise_lr, type = "response", newdata = ploan_val)
stepwise_target <- ploan_val$ploan_target
stepwise_predicted <- rep(0, length(stepwise_target))
stepwise_predicted[which(stepwise_response >= 0.5)] <- 1
cm_stepwise <- table(stepwise_target, stepwise_predicted)
perf_mat[4,] <- perf_eval(cm_stepwise)
```

R Exercise

- Evaluate the classification performances

  ✓ Confusion matrices

```
> cm_full
          full_predicted
full_target   0   1
          0 661   3
          1  34  52
```

```
> cm_forward
             forward_predicted
forward_target   0   1
             0 663   1
             1  50  36
```

```
> cm_backward
              backward_predicted
backward_target   0   1
              0 661   3
              1  34  52
```

```
> cm_stepwise
              stepwise_predicted
stepwise_target   0   1
              0 663   1
              1  50  36
```

  ✓ Classification performances

```
> perf_mat
                  TPR Precision       TNR       ACC       BCR        F1
LR_ALL      0.6046512 0.9454545 0.9954819 0.9506667 0.7758346 0.7375887
LR_Forward  0.4186047 0.9729730 0.9984940 0.9320000 0.6465093 0.5853659
LR_Backward 0.6046512 0.9454545 0.9954819 0.9506667 0.7758346 0.7375887
LR_Stepwise 0.4186047 0.9729730 0.9984940 0.9320000 0.6465093 0.5853659
```