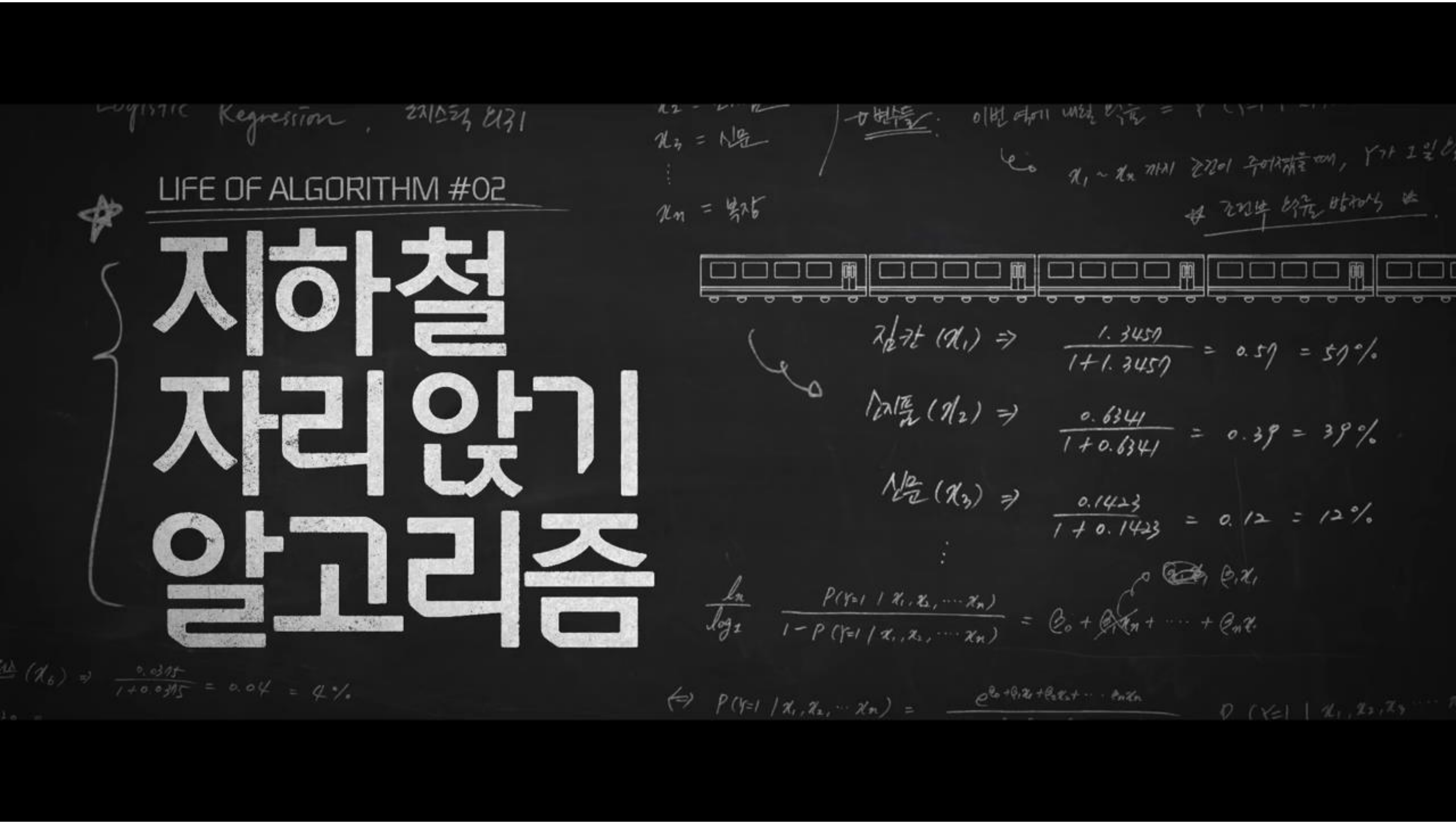
Lecture 5: Logistic Regression

Pilsung Kang
School of Industrial Management Engineering
Korea University

AGENDA

Logistic Regression: Intro.



Logistic Regression

- Classification

Men

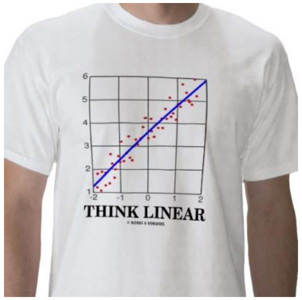
Vs.

Women

Revisit Multiple Linear Regression

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

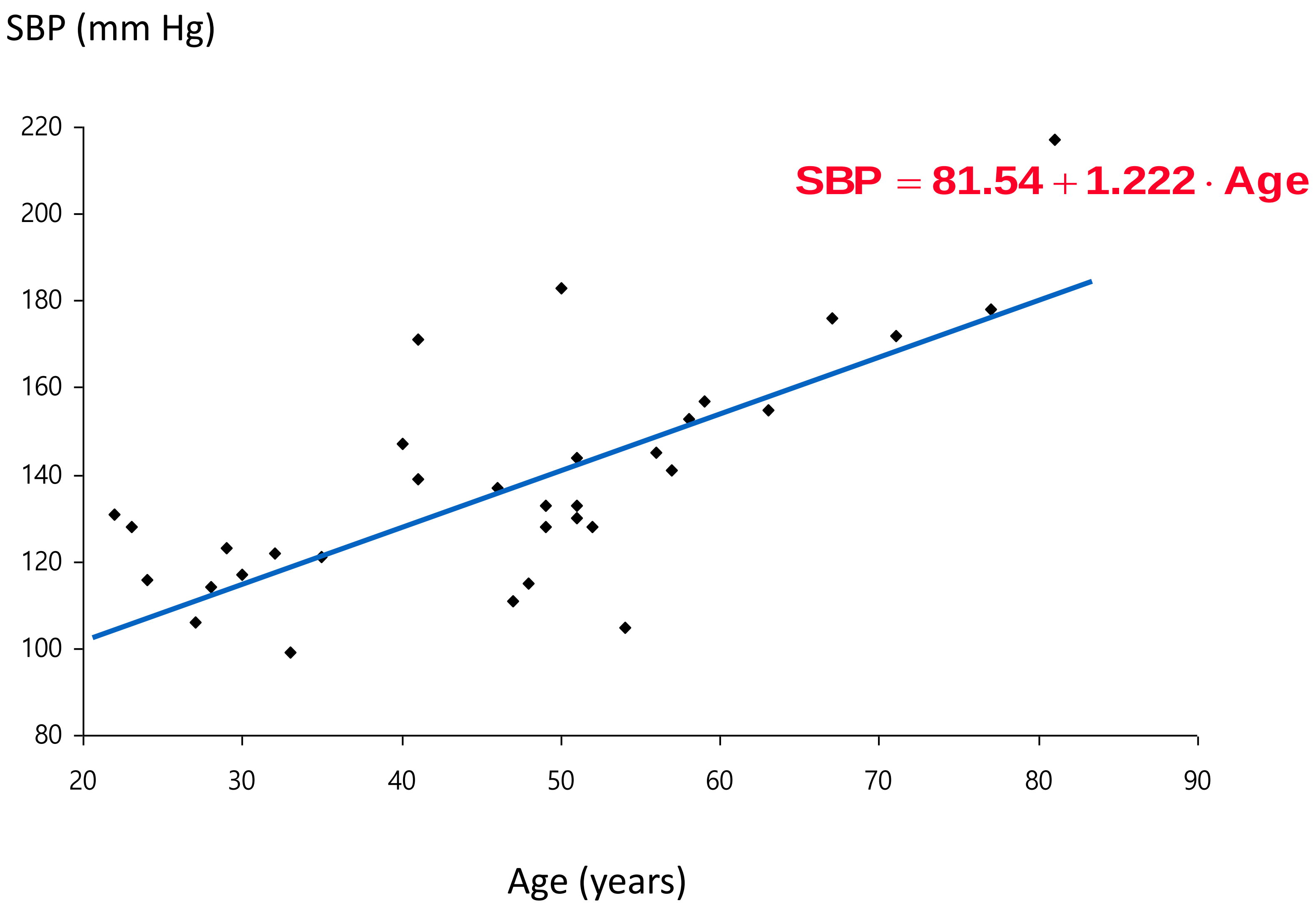
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$



- 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



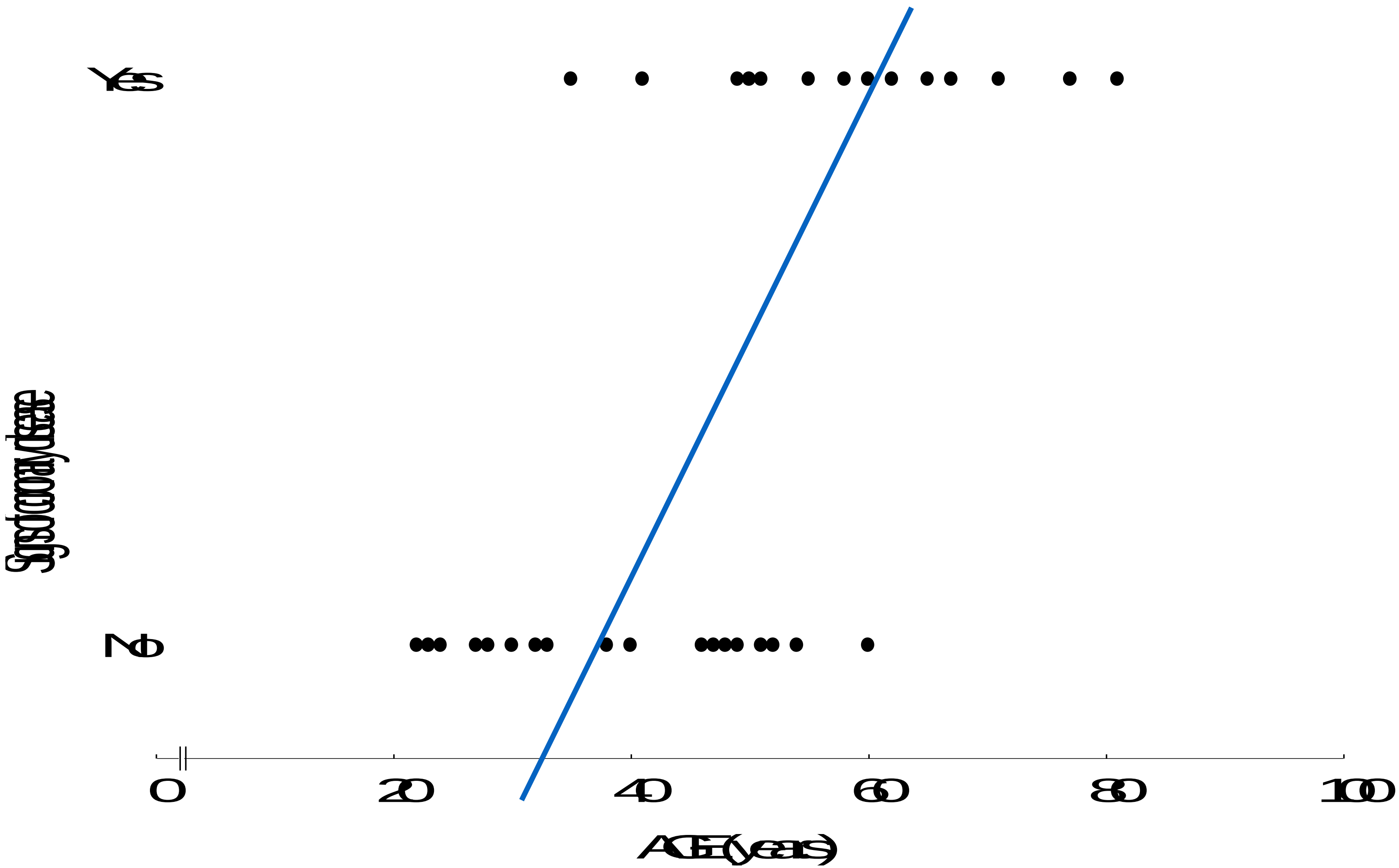
What If

- 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

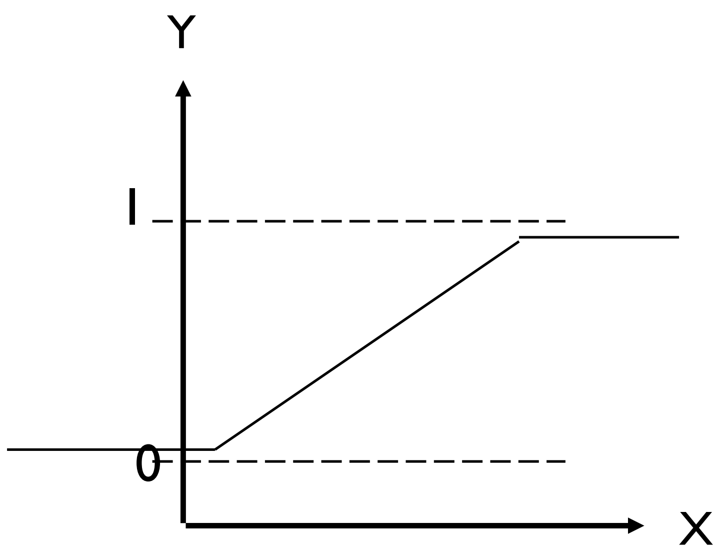
- Linear regression does not estimate Pr(Y=1|X) well



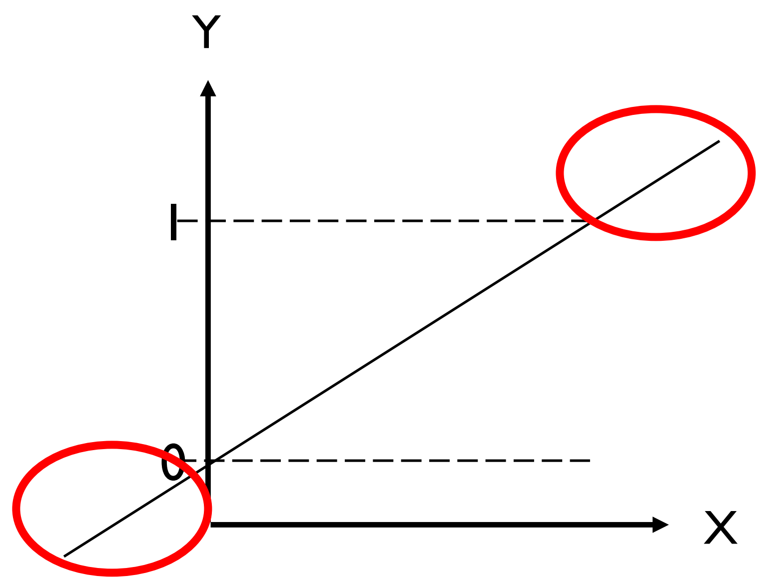
For Classification Task

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

Ideally:

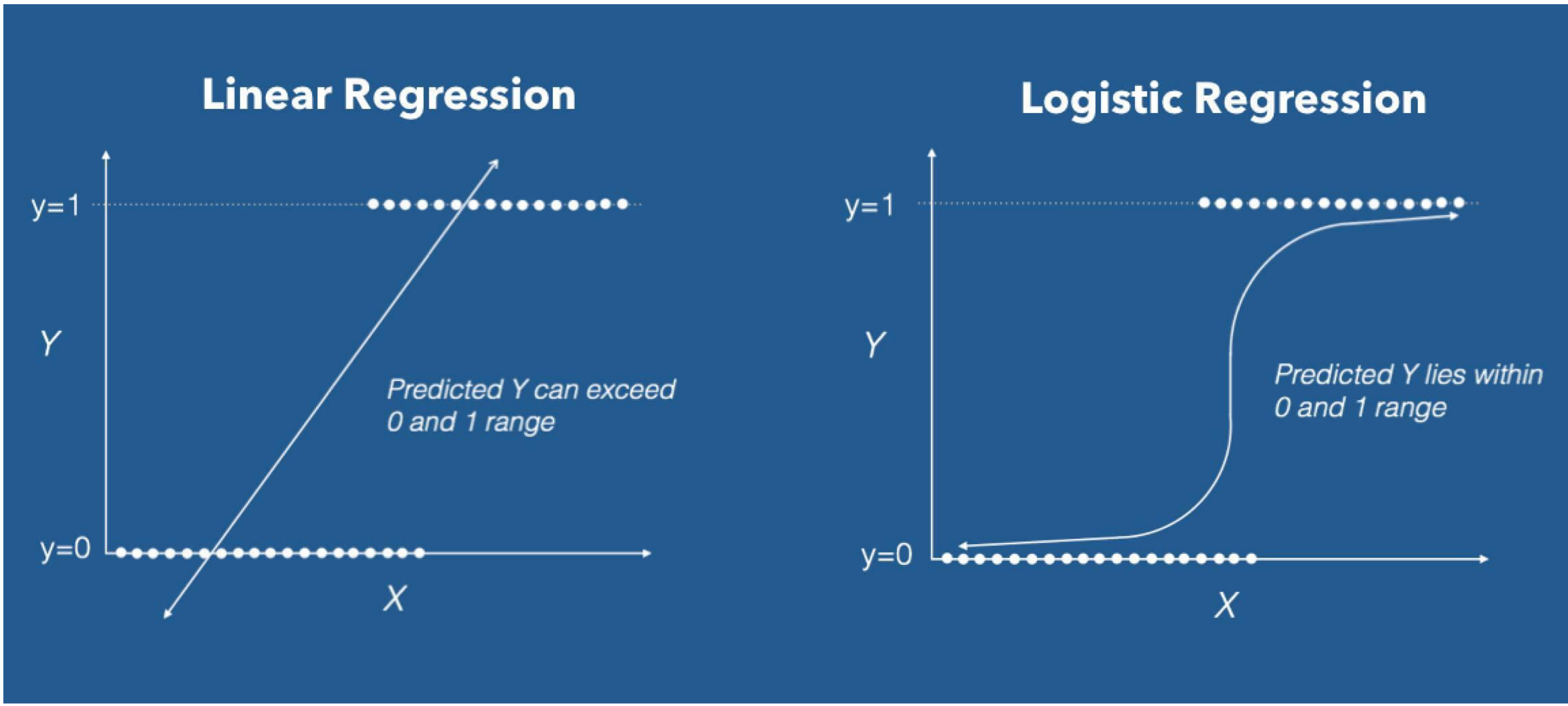


Reality:



For Classification Task

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



https://medium.com/greyatom/logistic-regression-89e496433063

For Classification

- Problem
  - ✓ For binary classification tasks, there only two possible outcomes (0 and 1)
  - ✓ Regression equation has no limit on the generated value
  - ✓ Allowed ranges of the input X and the output y do not match

$$\hat{y} = \boxed{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d}$$

Only 0 or 1 are allowed

All real values are possible

  - ✓ Goal: Build a classification model that inherit the advantages of regression model (ability to find significant variables, explainability, etc)

Logistic Regression

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

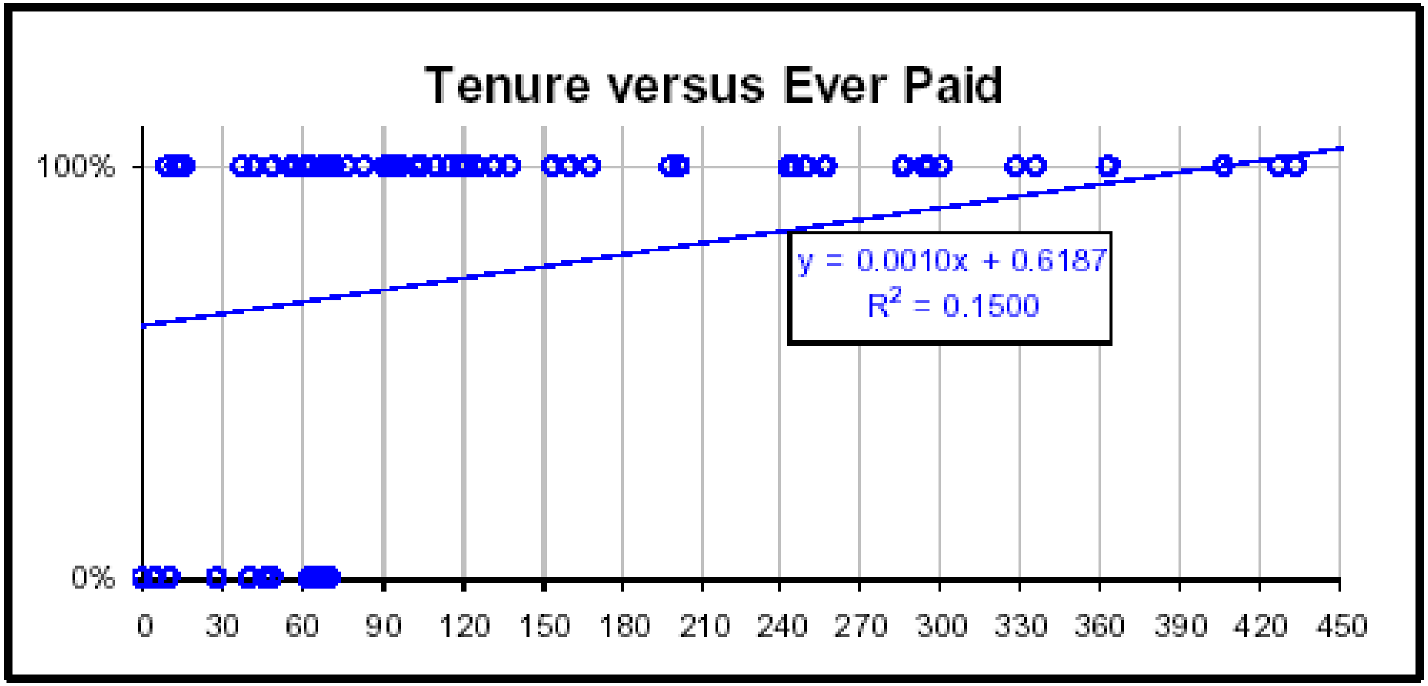
- Features:
  - ✓ Instead of Y as outcome variable (like in linear regression), we use a function of Y called the "logit".
  - ✓ Logit can be modeled as a linear function of the predictors.
  - ✓ The logit can be mapped back to a probability, which, in turn, can be mapped to a class.

For Classification Task

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

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

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



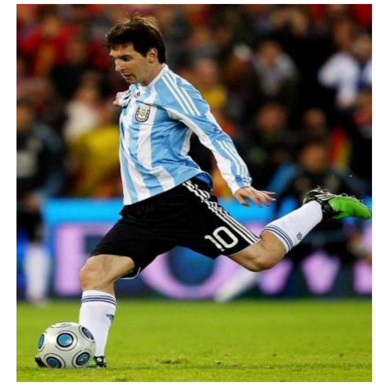
Logistic Regression: Odds

- 2010 World Cup Betting Odds

9 : 2

9 : 2

6 : 1



9 : 1

200 : 1

250 : 1

500 : 1

1000 : 1

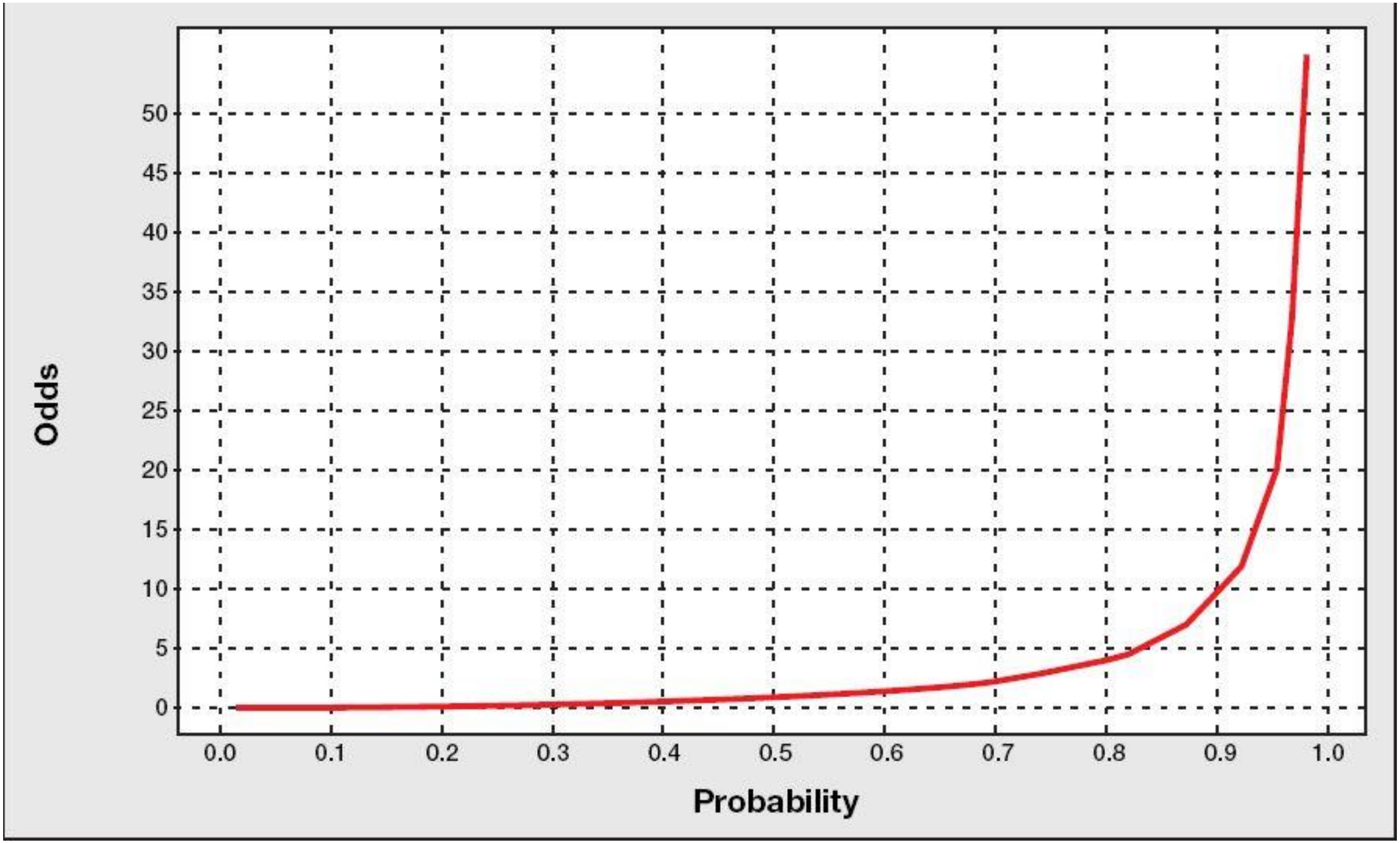
Logistic Regression: Odds

- 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

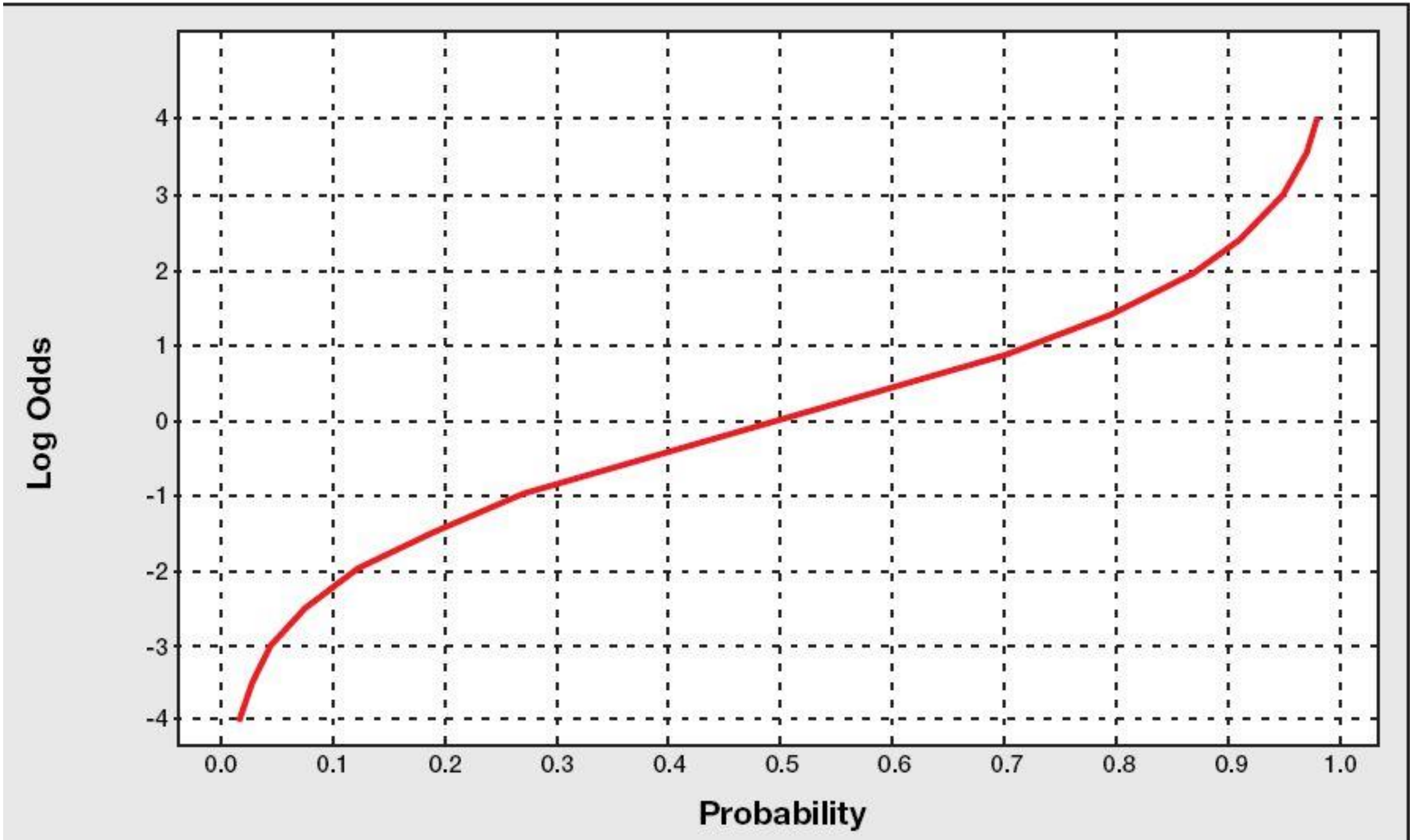
- The limitation of the odds
  - ✓ $0 < odds < \infty$
  - ✓ Asymmetric

- Take the logarithm of the odds

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

  - ✓ $-\infty < \log(odds) < \infty$
  - ✓ 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:

$$log(Odds) = log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

  ✓ Take the exponential for the both sides:

$$\frac{p}{1-p} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d}$$

  ✓ For the probability of the success:

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}} = \boxed{\sigma(\mathbf{x}|\beta)}$$

Logistic Regression: Learning

- Estimating the coefficients
  - ✓ Assume that we have two different logistic models, each of which makes the predictions for the same dataset as below, which model is better?

Model A

Glass	Label	P(Y=1)	P(Y=0)
1	1	0.908	0.092
2	0	0.201	0.799
3	1	0.708	0.292
4	0	0.214	0.786
5	1	0.955	0.045
6	0	0.017	0.983
7	1	0.807	0.193
8	0	0.126	0.874
9	1	0.937	0.063
10	0	0.068	0.932

Model B

Glass	Label	P(Y=1)	P(Y=0)
1	1	0.557	0.443
2	0	0.425	0.575
3	1	0.604	0.396
4	0	0.387	0.613
5	1	0.615	0.385
6	0	0.356	0.644
7	1	0.406	0.594
8	0	0.508	0.492
9	1	0.704	0.296
10	0	0.325	0.675

  - ✓ Model A is better than Model B because Model A generates higher probabilities for the actual labels

Logistic Regression: Learning

- Estimating the coefficients
  - ✓ Likelihood function
    - Likelihood for an individual object is its predicted probability being classified as the correct class
      - Likelihood of Glass 1 is 0.908
      - Likelihood of Glass 2 is 0.799
    - If the objects are assumed to be generated independently, the likelihood of the entire dataset is the product of every object's likelihood
    - Generally the likelihood of a dataset is very small (values between 0 and 1 are compounded), log-likelihood is commonly used

Model A

Glass	Label	P(Y=1)	P(Y=0)
1	1	0.908	0.092
2	0	0.201	0.799
3	1	0.708	0.292
4	0	0.214	0.786
5	1	0.955	0.045
6	0	0.017	0.983
7	1	0.807	0.193
8	0	0.126	0.874
9	1	0.937	0.063
10	0	0.068	0.932

Logistic Regression: Learning

- Estimating the coefficients
  - ✓ Likelihood function

Model A

Glass	Label	P(Y=1)	P(Y=0)	우도	로그 우도
1	1	0.908	0.092	0.908	-0.0965
2	0	0.201	0.799	0.799	-0.2244
3	1	0.708	0.292	0.708	-0.3453
4	0	0.214	0.786	0.786	-0.2408
5	1	0.955	0.045	0.955	-0.0460
6	0	0.017	0.983	0.983	-0.0171
7	1	0.807	0.193	0.807	-0.2144
8	0	0.126	0.874	0.874	-0.1347
9	1	0.937	0.063	0.937	-0.0651
10	0	0.068	0.932	0.932	-0.0704
				0.233446	**-0.1455**

Model B

Glass	Label	P(Y=1)	P(Y=0)	우도	로그 우도
1	1	0.557	0.443	0.557	-0.5852
2	0	0.425	0.575	0.575	-0.5534
3	1	0.604	0.396	0.604	-0.5042
4	0	0.387	0.613	0.613	-0.4894
5	1	0.615	0.385	0.615	-0.4861
6	0	0.356	0.644	0.644	-0.4401
7	1	0.406	0.594	0.406	-0.9014
8	0	0.508	0.492	0.492	-0.7093
9	1	0.704	0.296	0.704	-0.3510
10	0	0.325	0.675	0.675	-0.3930
				0.004458	**-0.5413**

- ✓ Model A's (log) likelihood is greater than that of Model B
- ✓ Model A can explain the dataset better than Model A

Logistic Regression: Learning

- Maximum likelihood estimation (MLE)
  - ✓ Find the coefficients that maximizes the likelihood of the dataset
  - ✓ Likelihood of the object i

$$P(\mathbf{x}_i, y_i|\boldsymbol{\beta}) = \begin{cases} \sigma(\mathbf{x}_i|\boldsymbol{\beta}), & if \quad y_i = 1 \\ 1 - \sigma(\mathbf{x}_i|\boldsymbol{\beta}), & if \quad y_i = 0 \end{cases}$$

  - ✓ Since the y_i is either 0 or 1, we can rewrite the above probability as follows:

$$P(\mathbf{x}_i, y_i|\boldsymbol{\beta}) = \sigma(\mathbf{x}_i|\boldsymbol{\beta})^{y_i}(1 - \sigma(\mathbf{x}_i|\boldsymbol{\beta}))^{1-y_i}$$

Logistic Regression: Learning

- Maximum likelihood estimation (MLE)
  - ✓ Assume that the objects are independently generated, the likelihood of the entire dataset is expressed as follows:

$$L(\mathbf{X}, \mathbf{y}|\boldsymbol{\beta}) = \prod_{i=1}^{N} P(\mathbf{x}_i, y_i|\boldsymbol{\beta}) = \prod_{i=1}^{N} \sigma(\mathbf{x}_i|\boldsymbol{\beta})^{y_i} (1 - \sigma(\mathbf{x}_i|\boldsymbol{\beta}))^{1-y_i}$$

  - ✓ Take a log for the both sides,

$$logL(\mathbf{X}, \mathbf{y}|\boldsymbol{\beta}) = \sum_{i=1}^{N} y_i \sigma(\mathbf{x}_i|\boldsymbol{\beta}) + (1 - y_i)(1 - \sigma(\mathbf{x}_i|\boldsymbol{\beta}))$$

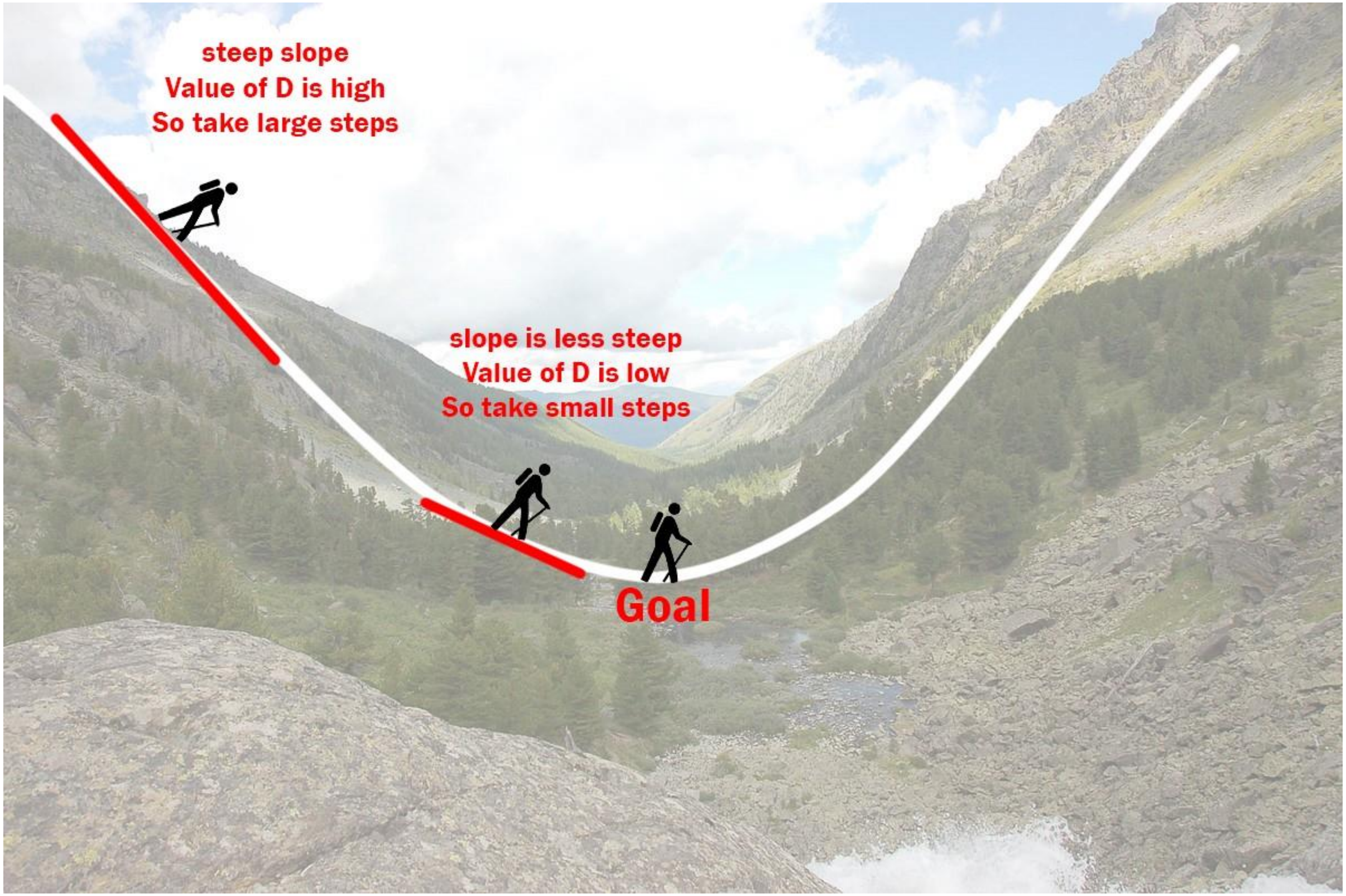
  - ✓ (Log) likelihood is non-linear with β, there is no explicit solution as in MLR
    - Find the solution with an optimization algorithm such as Gradient Descent

Logistic Regression: Learning

- Gradient Descent

Logistic Regression: Learning

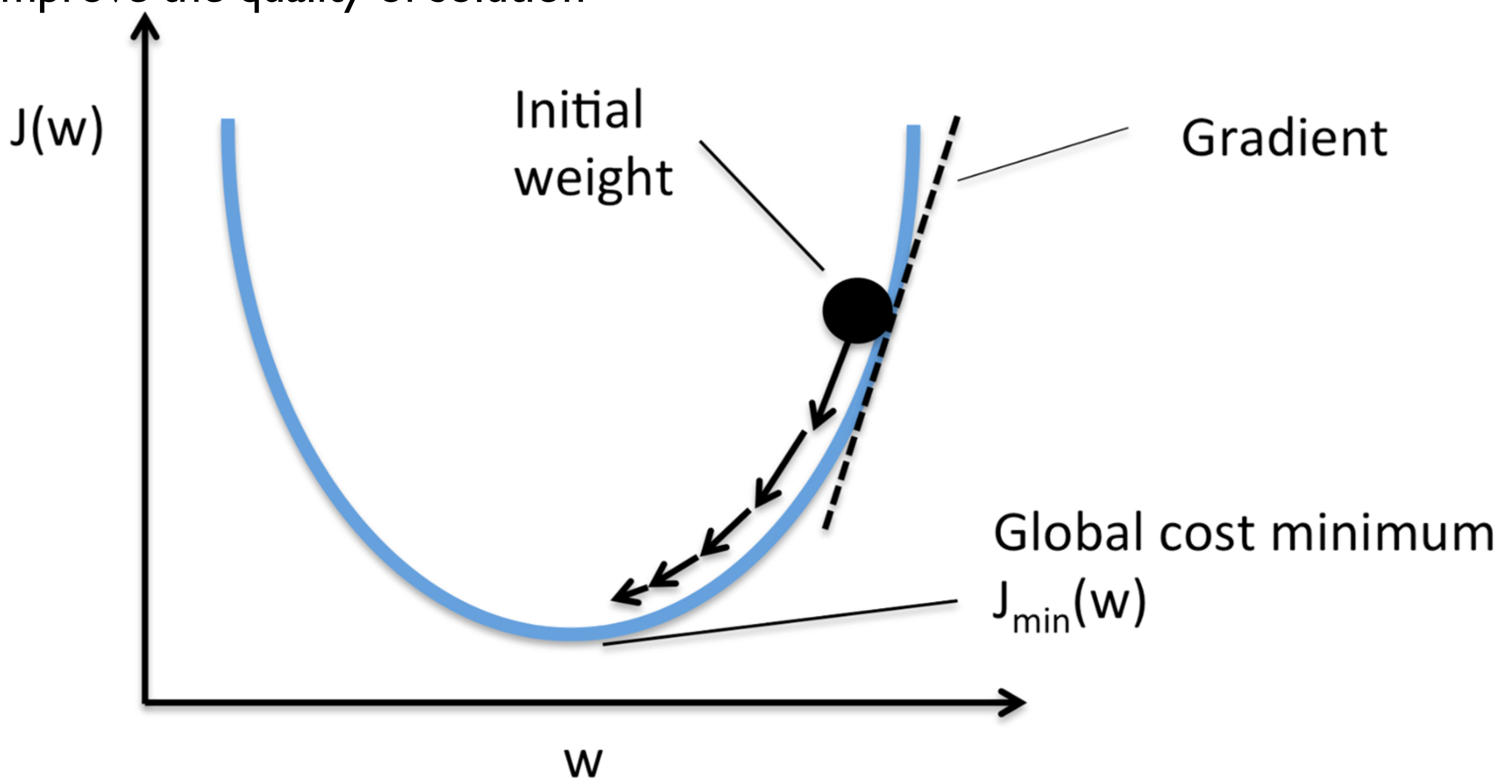
- Gradient Descent



https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931

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

Gradient Descent Algorithm

- Take the first derivative of the cost function w.r.t the current weight w
  - ✓ Is the gradient 0?
    - Yes: Current weights are the optimum! → end of learning
    - No: Current weights can be improved → learn more
  - ✓ How can we improve the current weights if the gradient is not 0?
    - Move the current weight toward to the opposite direction of the gradient
  - ✓ How much should the weights be moved?
    - Not sure
    - Move them a little and compute the gradient again
    - It will converge

Logistic Regression: Learning

- Theoretical Background (Optional)
  - ✓ Taylor expansion

$$f(w + \Delta w) = f(w) + \frac{f'(w)}{1!}\Delta w + \frac{f''(w)}{2!}(\Delta w)^2 + \cdots$$

  - ✓ If the first derivative is not zero, we can decrease the function value by moving x toward the opposite direction of its first derivative

Which direction to go?

$$w_{new} = w_{old} - \alpha f'(w), \quad \text{where } 0 < \alpha < 1.$$

How far should we move?

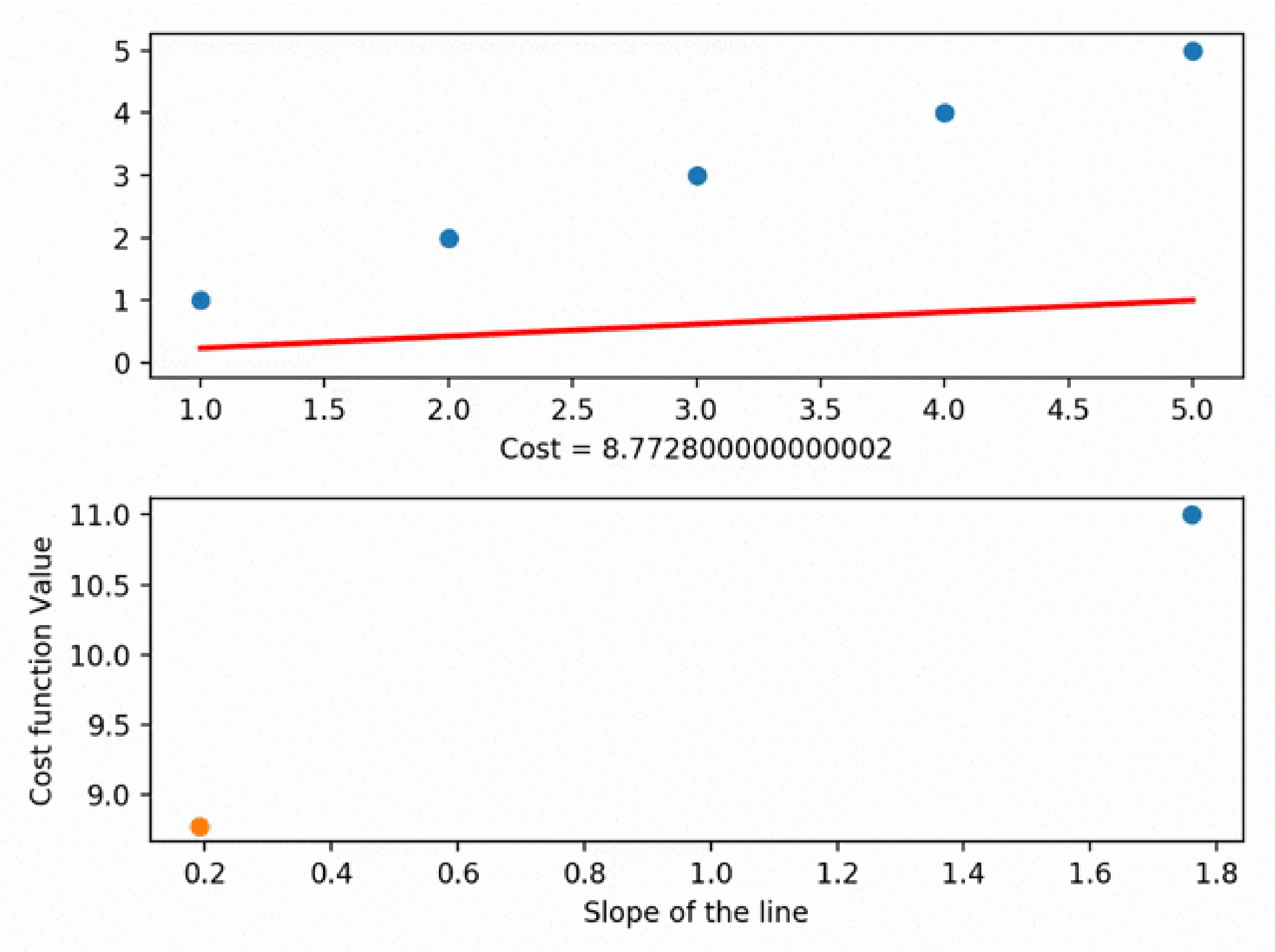
  - ✓ Then the function value of the new x is always smaller than that of the old x

$$f(w_{new}) = f(w_{old} - \alpha f'(w_{old})) \cong f(w_{old}) - \alpha \left| f'(w) \right|^2 < f(w_{old})$$

https://www.youtube.com/watch?v=3d6DsjIBzJ4&t=322s

Logistic Regression: Learning

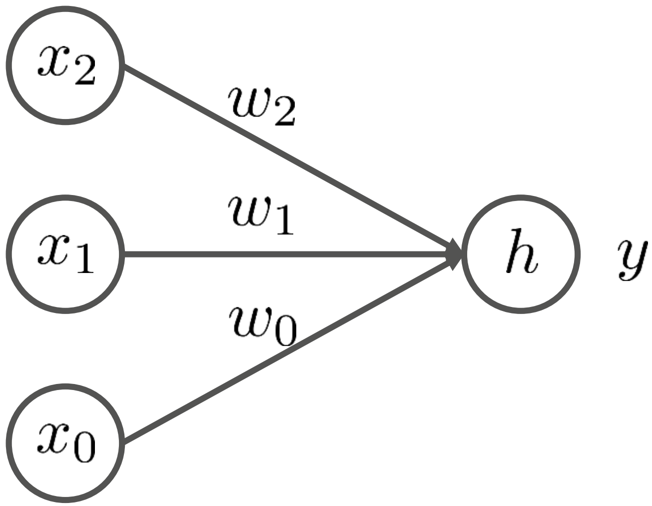
- Illustrative example



https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b

Logistic Regression: Learning

- Gradient descent with two input variables



$$h = \sum_{i=0}^{2} w_i x_i$$

$$y = \frac{1}{1 + exp(-h)}$$

- Let's define the squared loss function $L = \frac{1}{2}(t - y)^2$
- How to find the gradient w.r.t. w or x?

Logistic Regression: Learning

- Use chain rule

$$\frac{\partial L}{\partial y} = y - t$$

$$\frac{\partial y}{\partial h} = \frac{exp(-h)}{(1 + exp(-h))^2} = \frac{1}{1 + exp(-h)} \cdot \frac{exp(-h)}{1 + exp(-h)} = y(1 - y)$$

$$\frac{\partial h}{\partial w_i} = x_i$$

- Gradients for w and x

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial w_i} = (y - t) \cdot y(1 - y) \cdot x_i$$

- Update w

$$w_{new} = w_{old} - \alpha \times \frac{L}{\partial w_i} = w_{old} - \alpha \times (y - t) \cdot y(1 - y) \cdot x_i$$

Logistic Regression: Learning

- Weight update by Gradient Descent

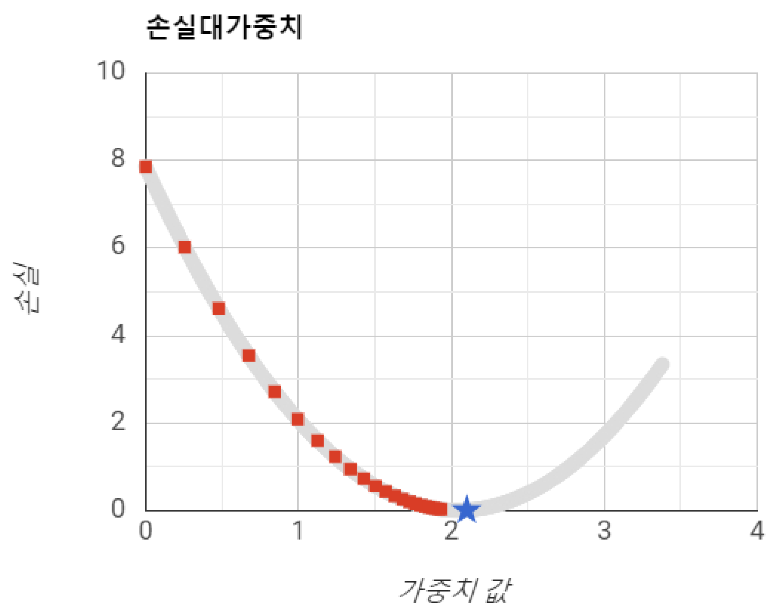
$$w_i^{new} = w_i^{old} - \alpha \times (y - t) \cdot 1 \cdot h(1 - h) \cdot x_i$$

Update the coefficient more if the current output y is very different from the target t

Update the coefficients more if the value of corresponding input variable is large

Logistic Regression: Learning

- The Effect of learning rate α

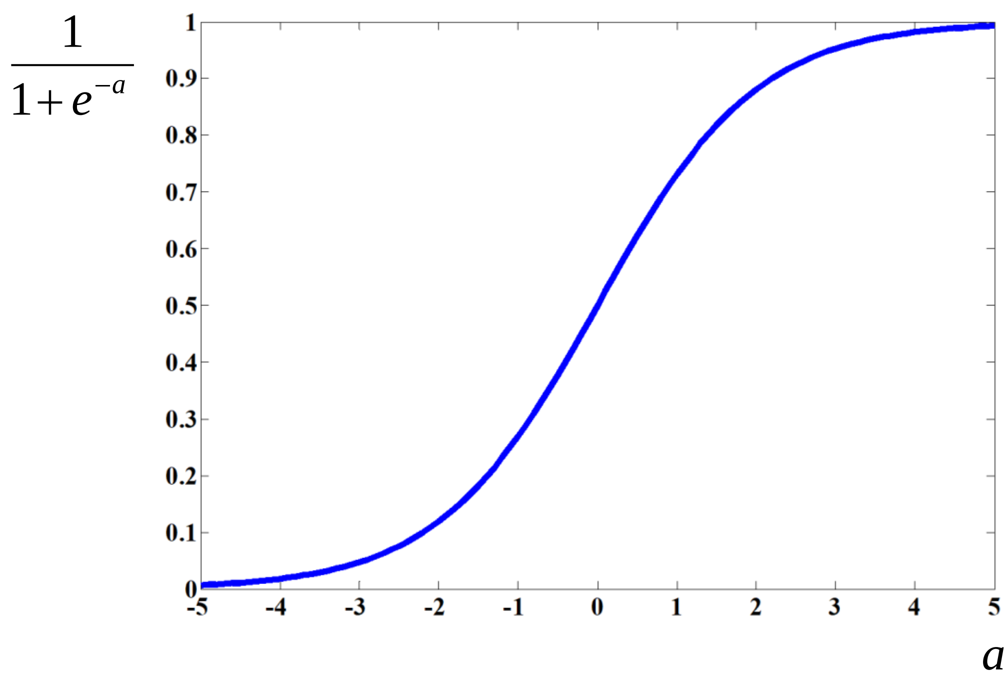


https://developers.google.com/machine-learning/crash-course/fitter/graph

Logistic Regression: Prediction

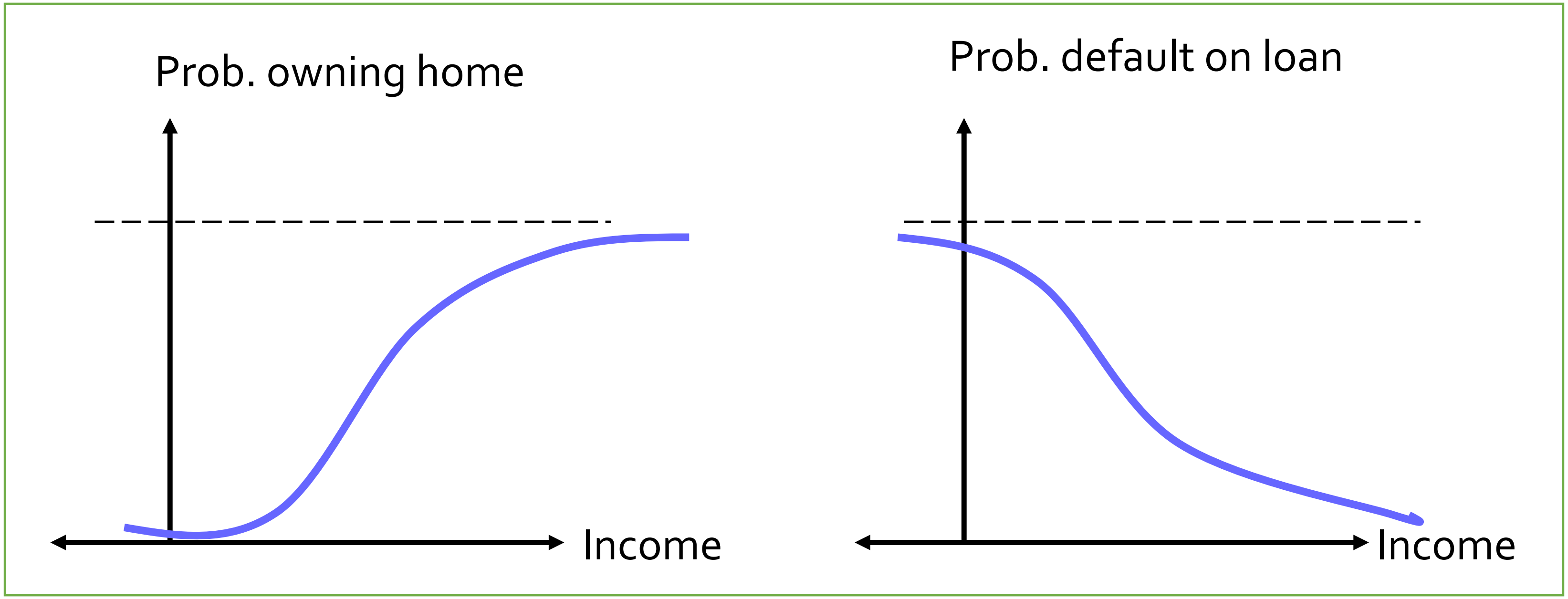
- Success probability
  - ✓ When a set of predictors (independent variables) are given, we can estimate the probability of the success.

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$



For Classification Task

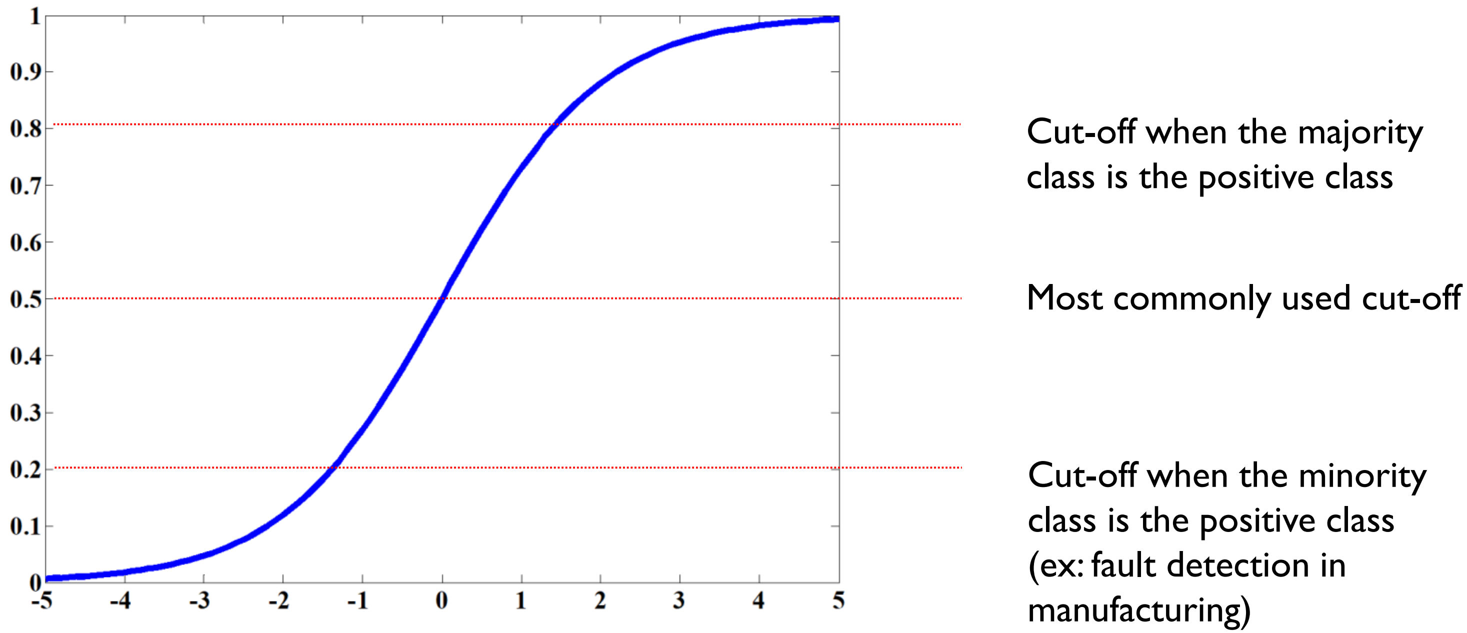
- In real cases…
  - ✓ The probability may follow a certain type of curve rather than a straight line.



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: Interpretation

- Meaning of coefficients
  - ✓ Linear regression

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

    - The amount of target variable changes when the input variable is increased by 1
  - ✓ Logistic regression

$$log(Odds) = log\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

    - The amount of log odd changes when the input variable is increased by 1 (not intuitive)

Logistic Regression: Interpretation

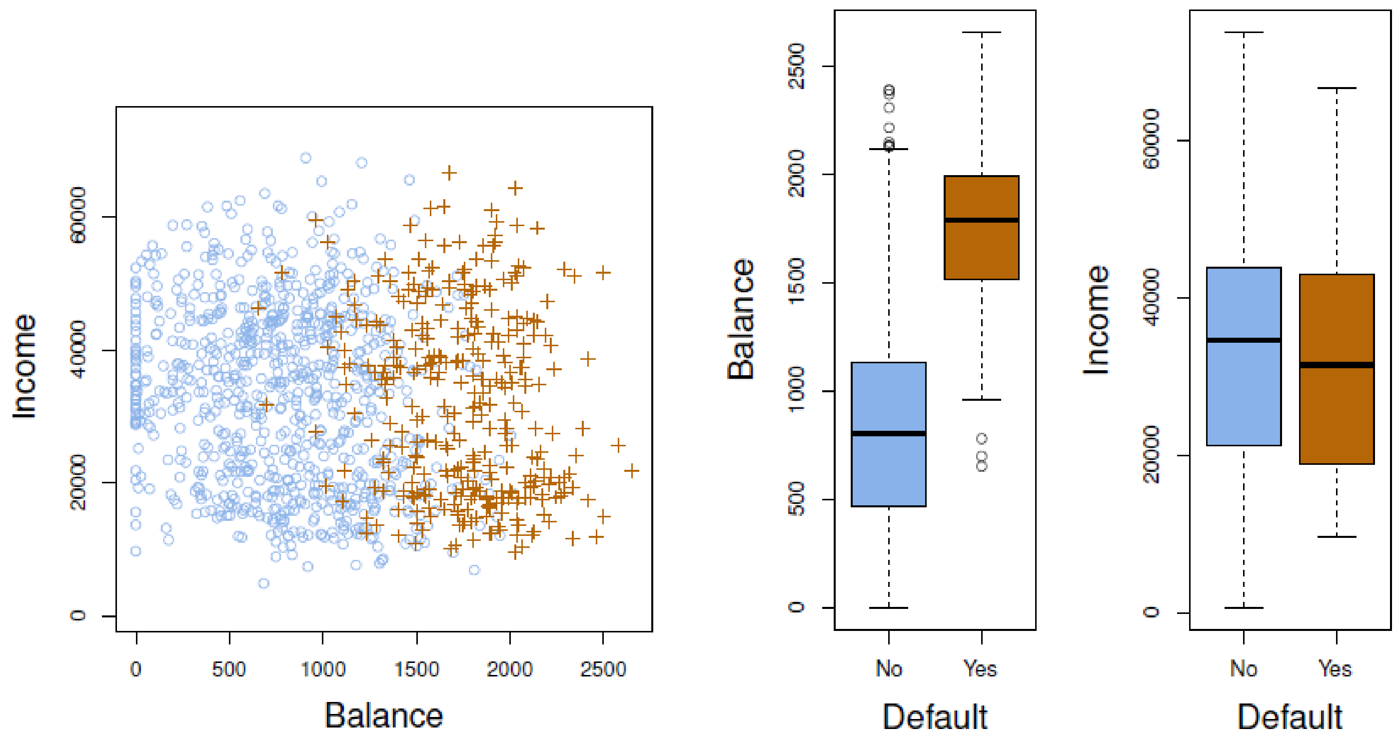
- Odds ratio
  - ✓ Suppose that the value of x_1 is increased by one unit from x_1 to x_1+1, while the other predictors are held at their current value.
  - ✓ Odds ratio:

$$\frac{odds(x_1 + 1, \cdots, x_d)}{odds(x_1, \cdots, x_d)} = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 (x_1 + 1) + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_d x_d}}{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_d x_d}} = e^{\hat{\beta}_1}$$

  - ✓ When x_1 is increased by 1, then the odds is increased(decreased) by a factor of $e^{\hat{\beta}_1}$
    - Coefficient is positive → success probability increases when the corresponding input value increases (success class and coefficient are positively correlated)
    - Coefficient is positive → success probability increases when the corresponding input value increases (success class and coefficient are negatively correlated)

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

$$p = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d)}}$$

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

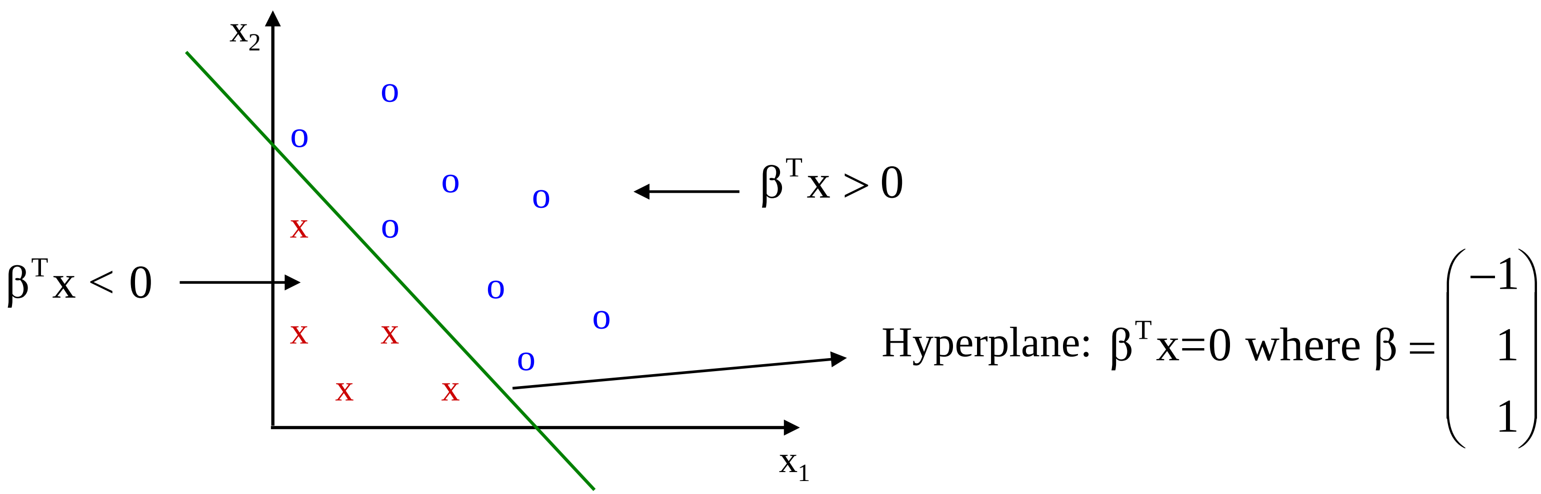
Logistic Regression: Interpretation

- 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.



Classifier

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

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

Logistic Regression: Interpretation

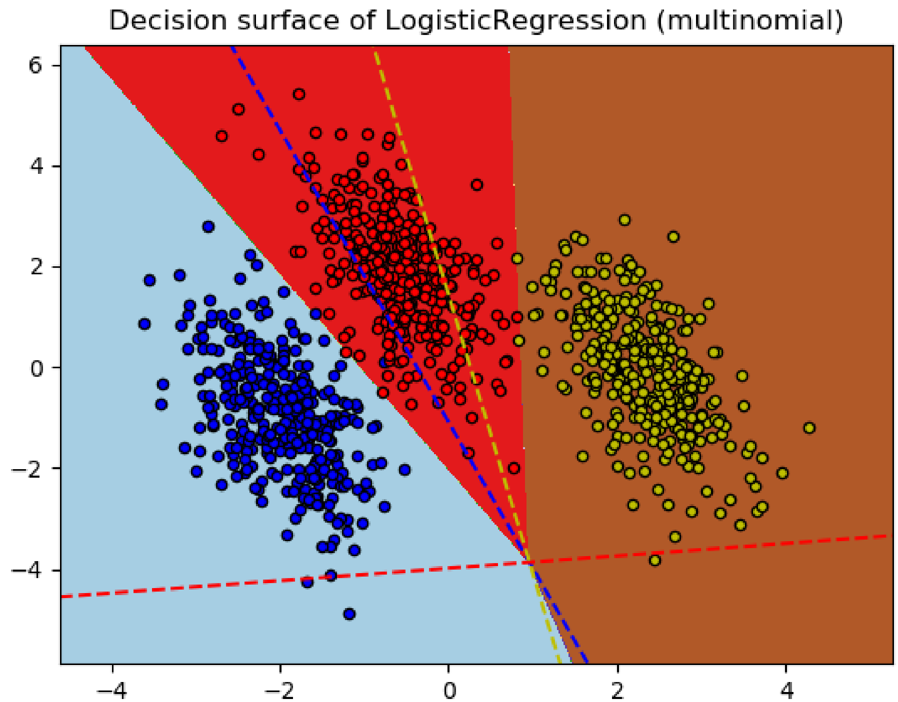
- Profiling
  - ✓ Finding factors that differentiate between the two classes.
  - ✓ After variable selection:

$$\frac{p}{1-p} = e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d}$$

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

Multinomial Logistic Regression

- Basic Logistic Regression is developed to solve the binary classification problem
  - ✓ Q) Can we use the logistic regression to classify more than 3 classes?



http://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_multinomial.html

Multinomial Logistic Regression

- Multinomial logistic regression
  - ✓ Set the baseline class and formulate the regression equation for the relative log odds to this class
  - ✓ Ex) If there are three classes, estimate the coefficients of the following two regression models
    - Logistic regression of Class 1 versus Class 3

$$log\left(\frac{p(y=1)}{p(y=3)}\right) = \hat{\beta_{10}} + \hat{\beta_{11}}x_1 + \hat{\beta_{12}}x_2 \cdots + \hat{\beta_{1d}}x_d = \boldsymbol{\beta}_{1\cdot}^T\mathbf{x}$$

    - Logistic regression of Class 2 versus Class 2

$$log\left(\frac{p(y=2)}{p(y=3)}\right) = \hat{\beta_{20}} + \hat{\beta_{21}}x_1 + \hat{\beta_{22}}x_2 \cdots + \hat{\beta_{2d}}x_d = \boldsymbol{\beta}_{2\cdot}^T\mathbf{x}$$

Multinomial Logistic Regression

- Multinomial logistic regression
  - ✓ Why do we learn only two models although there are three classes? (Generally, why do we learn (K-1) models when there are K classes?)
    - For each object, the sum of likelihoods must be 1, so that if we know (K-1) likelihoods, that the rest can be automatically computed

$$\frac{p(y=1)}{p(y=3)} = e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} \qquad \frac{p(y=2)}{p(y=3)} = e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}}$$

$$p(y=1) + p(y=2) + p(y=3) = 1$$

$$p(y=3) \times e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} + p(y=3) \times e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}} + p(y=3) = 1$$

$$p(y=3) = \frac{1}{1 + e^{\boldsymbol{\beta}_{1\cdot}^T \mathbf{x}} + e^{\boldsymbol{\beta}_{2\cdot}^T \mathbf{x}}}$$

Multinomial Logistic Regression

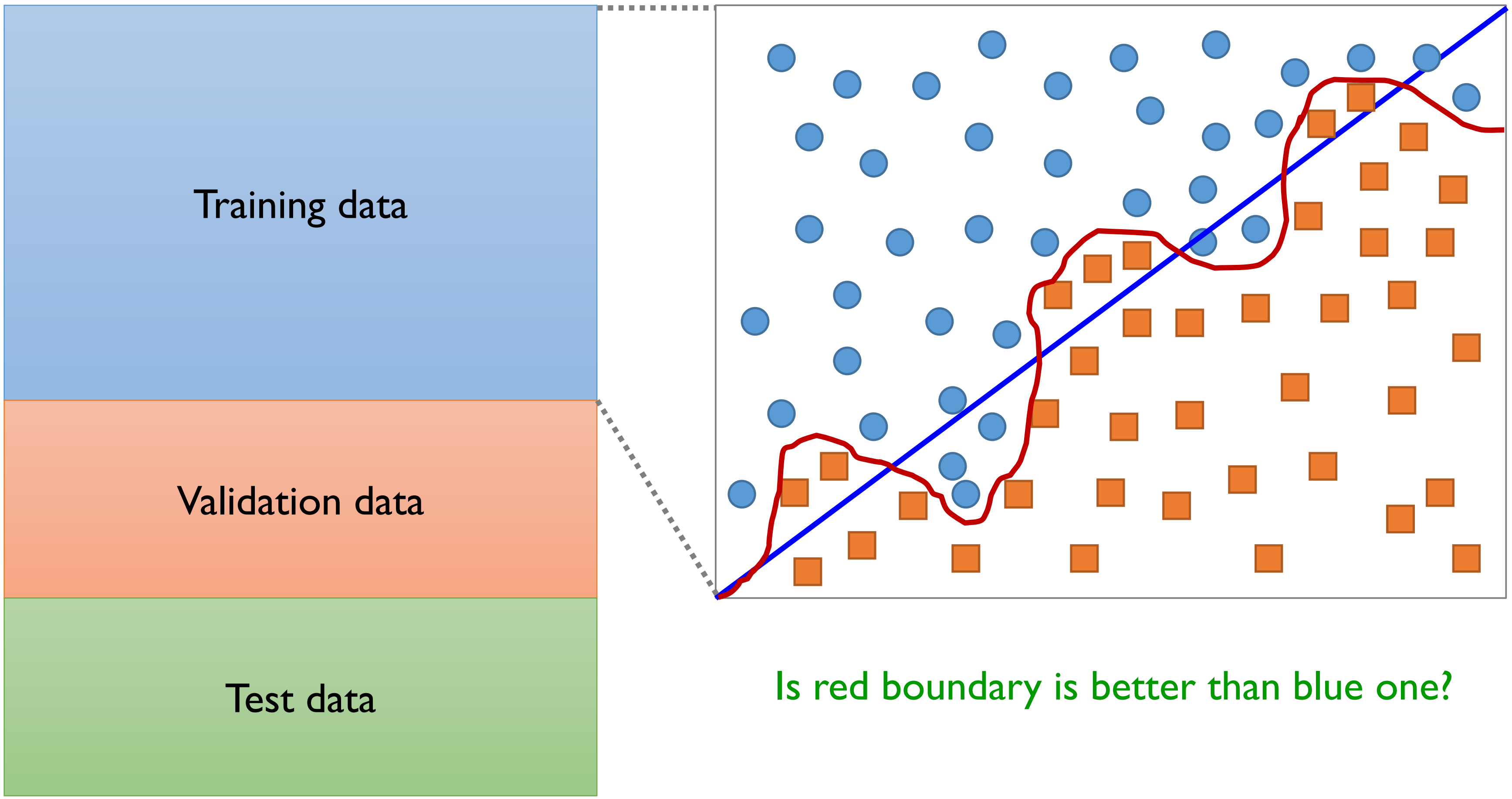
- Interpreting the coefficients in multinomial logistic regression
  - ✓ Interpret the coefficients for the two compared classes
    - Total phenols, Flavanoids, Monflavanoid penols, Hue, OD280~ variables are statistically significant for both 1 vs. 3, 2 vs. 3 models
    - Ash., Proanthocyanins variable is not statistically significant when discriminating the classes 1 and 3, but is significant when discriminating the classes 2 and 3

		1 vs 3		2 vs 3
	Coefficient	p-value	Coefficient	p-value
(Intercept)	-223.7894	0.0000	340.9326	0.0000
Alcohol.2	19.6193	0.7880	-35.2596	0.6828
Malic.acid.	1.0581	0.9228	-0.3022	0.9899
Ash.	14.6800	0.3881	-204.7437	0.0000
Alcalinity.of.ash.	-20.3881	0.8815	-2.2832	0.9864
Magnesium.	2.0553	0.9975	2.1132	0.9974
Total.phenols.	-169.4205	0.0000	-40.3325	0.0000
Flavanoids.	193.7935	0.0000	16.2013	0.0188
Nonflavanoid.phenols	93.5409	0.0000	214.1837	0.0000
Proanthocyanins.	15.5178	0.1453	115.3184	0.0000
Color.intensity.	-16.6775	0.4212	-11.5066	0.7671
Hue	-50.0008	0.0000	352.7617	0.0000
OD280.OD315.of.diluted.wines.	75.2435	0.0000	84.2914	0.0000
Proline.	-0.0120	1.0000	-0.2899	0.9999

AGENDA

Why Evaluate?

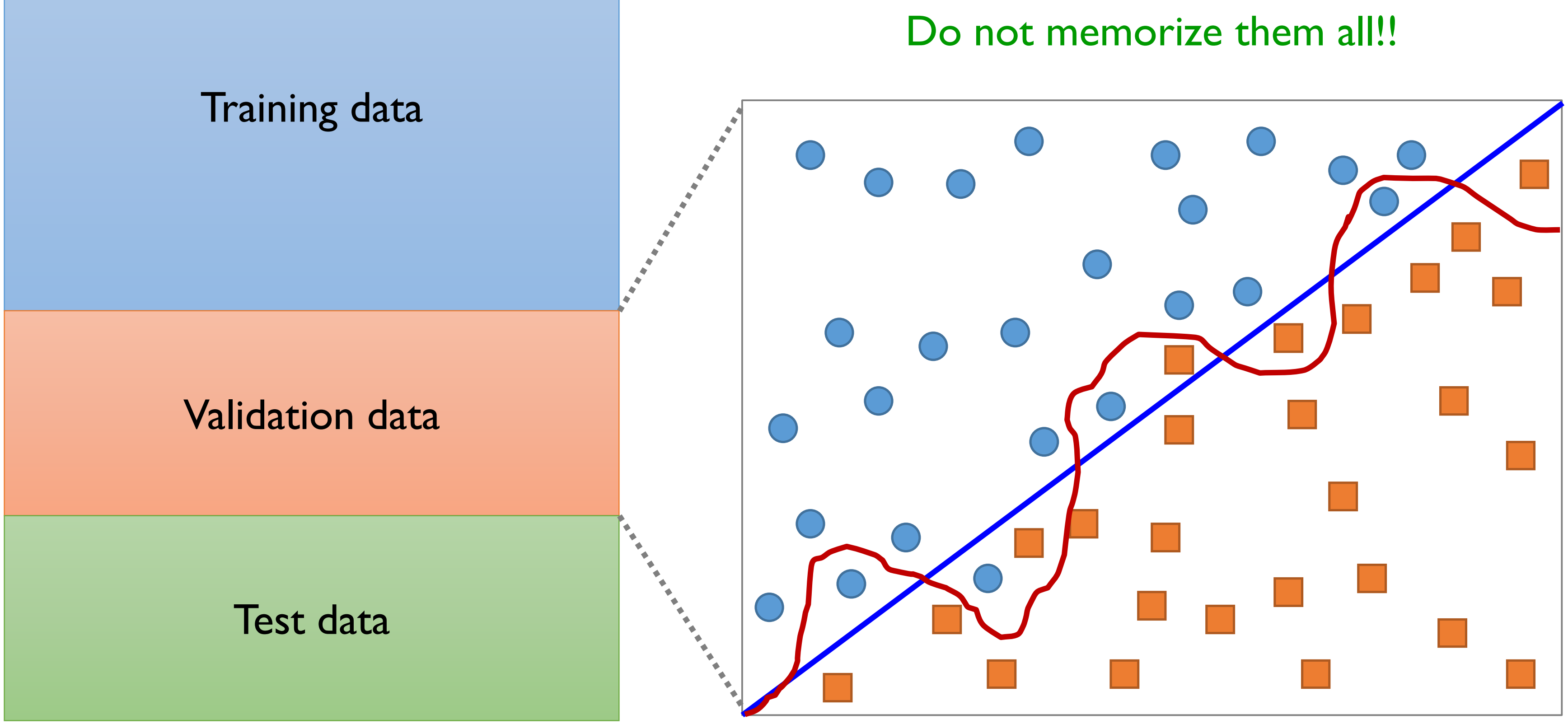
- Over-fitting for training data



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

I

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

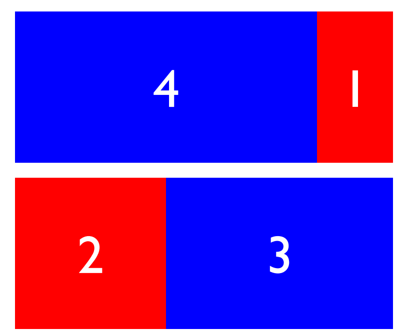
Confusion Matrix

- 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	M
Actual	F	4	1
	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 θ = 24,

Confusion Matrix		Predicted	
		F	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)
  - ✓ Fault Detection Problem:
    - Classify Good/Faulty products
    - A total of 100 products
    - 20 products are fault (Fault ratio: 0.2)
    - Label: 1(NG), 0(G)

Classification Performance

- Estimated likelihood (P(NG)) and the target label information

Glass	P(NG)	Label	Glass	P(NG)	Label	Glass	P(NG)	Label	Glass	P(NG)	Label
1	0.976	1	26	0.716	1	51	0.41	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	0.176	0
5	0.937	0	30	0.647	0	55	0.362	0	80	0.173	0
6	0.936	1	31	0.64	1	56	0.355	0	81	0.17	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	0.12	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	0.862	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	s0.071	0
15	0.847	1	40	0.548	0	65	0.273	0	90	0.069	0
16	0.845	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	0.51	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	0.01	0
24	0.741	0	49	0.434	0	74	0.2	0	99	0.005	0
25	0.737	0	50	0.414	0	75	0.193	0	100	0.002	0

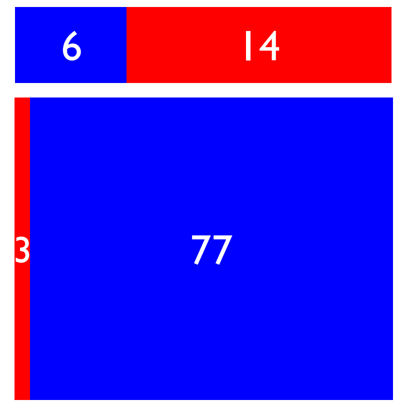
Classification Performance

4

Confusion matrix

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

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



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

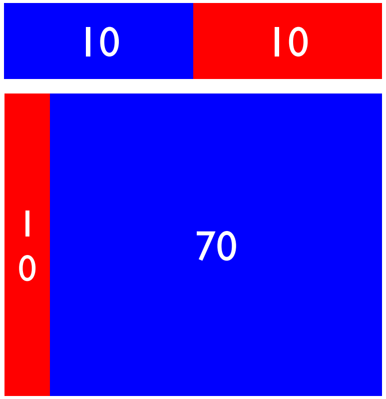
Classification Performance

4

Confusion matrix

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

Confusion Matrix		Predicted	
		M	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.

5

Classification Performance

ROC example

- First cut-off

Glass	P(NG)	Label
1	0.976	1
2	0.973	1
3	0.971	0
4	0.967	1
5	0.937	0
⋮	⋮	⋮

Confusion Matrix		예측	
		NG	G
실제	NG	0	20
	G	0	80

$$\mathrm{TPR} = \frac{0}{20} = 0$$

$$\mathrm{FPR} = \frac{0}{80} = 0$$

5

Classification Performance

ROC example

- Second cut-off

Glass	P(NG)	Label	TPR	FPR
			0	0
1	0.976	1		
2	0.973	1		
3	0.971	0		
4	0.967	1		
5	0.937	0		
⋮	⋮	⋮	⋮	⋮

Confusion Matrix		예측	
		NG	G
실제	NG	1	19
	G	0	80

$$\mathrm{TPR} = \frac{1}{20} = 0.05$$

$$\mathrm{FPR} = \frac{0}{80} = 0$$

5

Classification Performance

ROC example

- Third cut-off

Glass	P(NG)	Label	TPR	FPR
			0	0
1	0.976	1	0.05	0
2	0.973	1		
3	0.971	0		
4	0.967	1		
5	0.937	0		
⋮	⋮	⋮	⋮	⋮

Confusion Matrix		예측 NG	예측 G
실제	NG	2	18
실제	G	0	80

$$\mathrm{TPR} = \frac{2}{20} = 0.10$$

$$\mathrm{FPR} = \frac{0}{80} = 0$$

5

Classification Performance

ROC example

- Fourth cut-off

Glass	P(NG)	Label	TPR	FPR
			0.00	0.00
1	0.976	1	0.05	0.00
2	0.973	1	0.10	0.00
3	0.971	0		
4	0.967	1		
5	0.937	0		
⋮	⋮	⋮	⋮	⋮

Confusion Matrix		예측	
		NG	G
실제	NG	2	18
	G	1	79

$$\mathrm{TPR} = \frac{2}{20} = 0.10$$

$$\mathrm{FPR} = \frac{1}{80} = 0.0125$$

5

Classification Performance

ROC example

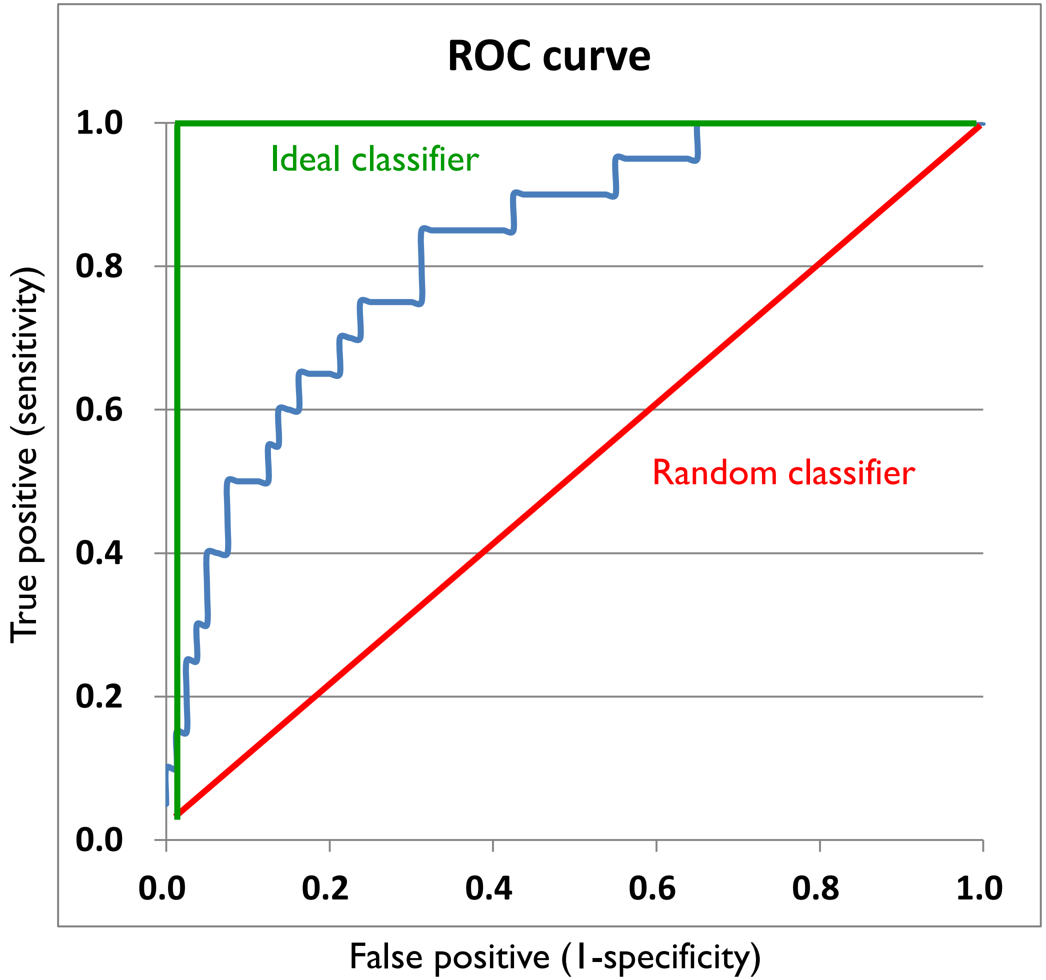
- Compute all possible TPR and FPR
- Draw a graph with FPR as an x-axis and TPR as an y-axis

Glass	P(NG)	Label	TPR	FPR
			0.000	0.000
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	0.01	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

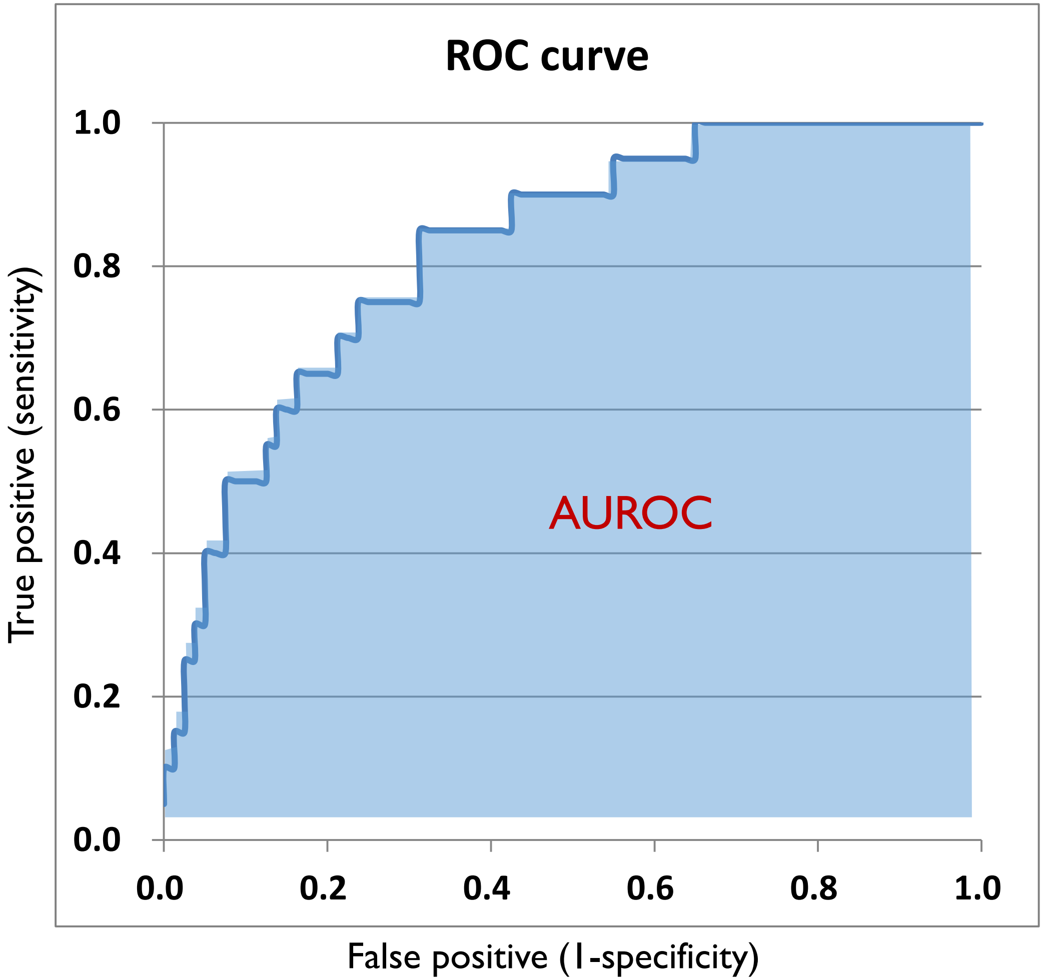


5

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 1: Binary Classification

- 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 1: Binary Classification

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

```
# Performance Evaluation Function -----------------------------------------
perf_eval2 <- 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 1: Binary Classification

- Initialize the performance matrix & Load the dataset

```r
# Initialize the performance matrix
perf_mat <- matrix(0, 1, 6)
colnames(perf_mat) <- c("TPR (Recall)", "Precision", "TNR", "ACC", "BCR", "F1")
rownames(perf_mat) <- "Logstic Regression"

# Load dataset
ploan <- read.csv("Personal Loan.csv")
input_idx <- c(2,3,4,6,7,8,9,11,12,13,14)
target_idx <- 10
ploan_input <- ploan[,input_idx]
ploan_target <- as.factor(ploan[,target_idx])
ploan_data <- data.frame(ploan_input, ploan_target)
```

✓ Column 1 & 5: id and zipcode (irrelevant variables)

✓ Column 10: target variable

✓ Convert the target variable type: numeric → factor

R Exercise 1: Binary Classification

- Normalize and split the dataset

```
# Conduct the normalization
ploan_input <- ploan[,input_idx]
ploan_input <- scale(ploan_input, center = TRUE, scale = TRUE)
ploan_target <- ploan[,target_idx]
ploan_data <- data.frame(ploan_input, ploan_target)

# Split the data into the training/validation sets
set.seed(12345)
trn_idx <- sample(1:nrow(ploan_data), round(0.7*nrow(ploan_data)))
ploan_trn <- ploan_data[trn_idx,] ploan_tst <- ploan_data[-trn_idx,]
```

✓ Conduct normalization for stable learning

✓ Divide the entire dataset into the training set (70%) and test set (30%)

R Exercise 1: Binary Classification

- Training the logistic regression model

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

  ✓ glm( ): generalized linear model

  - Arg 1: Formula
  - Arg 2: type of model (family = binomial → logistic regression)
  - Arg 3: training dataset

R Exercise 1: Binary Classification

- Training the logistic regression model

```
> summary(full_lr)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2973  -0.2366  -0.1081  -0.0482   3.6007

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.21016    0.22999 -18.306  < 2e-16 ***
Age                -0.05479    1.06837  -0.051  0.95910
Experience          0.23514    1.06214   0.221  0.82480
Income              2.07961    0.17125  12.144  < 2e-16 ***
Family              0.80944    0.13411   6.036 1.58e-09 ***
CCAvg               0.30738    0.10800   2.846  0.00442 **
Education           1.13270    0.14325   7.907 2.63e-15 ***
Mortgage            0.07188    0.08685   0.828  0.40790
Securities.Account -0.44039    0.15266  -2.885  0.00392 **
CD.Account          0.94355    0.12160   7.760 8.52e-15 ***
Online             -0.13209    0.12191  -1.083  0.27859
CreditCard         -0.61753    0.15835  -3.900 9.63e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R Exercise 1: Binary Classification

- Test the model and evaluate the classification performance

```r
lr_response <- predict(full_lr, type = "response", newdata = ploan_tst)
lr_target <- ploan_tst$ploan_target
lr_predicted <- rep(0, length(lr_target))
lr_predicted[which(lr_response >= 0.5)] <- 1
cm_full <- table(lr_target, lr_predicted)
cm_full
```

✓ predict function

- type = "response": return the probability belonging to the positive (1) class
- Set the cut-off value to 0.5
- Compute the confusion matrix

```
> cm_full
         lr_predicted
lr_target   0   1
        0 667   4
        1  26  53
```

R Exercise 1: Binary Classification

- Test the model and evaluate the classification performance

```
perf_mat[1,] <- perf_eval2(cm_full)
perf_mat
```

```
> perf_mat
                    TPR (Recall) Precision       TNR  ACC       BCR        F1
Logstic Regression     0.6708861 0.9298246 0.9940387 0.96 0.8166313 0.7794118
```

✓ The 67% of actual loan users are correctly identified by the logistic regression model

✓ The 93% of customers being identified by the model are actual loan users

✓ The 99.4% of actual non-users are correctly identified by the model

✓ The 96% of customers are correctly identified

R Exercise 2: Multi-class Classification

- Dataset: Wine

Wine Data Set

Download: Data Folder, Data Set Description

Abstract: Using chemical analysis determine the origin of wines

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1087880

The attributes are (dontated by Riccardo Leardi,
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

R Exercise 2: Multi-class Classification

- Install package, initiate the performance evaluation function

```r
# Multinomial logistic regression
install.packages("nnet")
library(nnet)

perf_eval3 <- function(cm){
     # Simple accuracy
     ACC <- sum(diag(cm))/sum(cm)
     # ACC for each class
     A1 <- cm[1,1]/sum(cm[1,])
     A2 <- cm[2,2]/sum(cm[2,])
     A3 <- cm[3,3]/sum(cm[3,])
     BCR <- (A1*A2*A3)^(1/3)
     return(c(ACC, BCR))
}
```

R Exercise 2: Multi-class Classification

- Load dataset, set the baseline class, divide the dataset

```
wine <- read.csv("wine.csv")
# Define the baseline class
wine$Class <- as.factor(wine$Class)
wine$Class <- relevel(wine$Class, ref = "3")

trn_idx <- sample(1:nrow(wine), round(0.7*nrow(wine)))
wine_trn <- wine[trn_idx,]
wine_tst <- wine[-trn_idx,]
```

✓ Original type of Class variable is “int” → convert its type to “factor”

R Exercise 2: Multi-class Classification

- Train the models

```
# Train multinomial logistic regression
ml_logit <- multinom(Class ~ ., data = wine_trn)

# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

✓ summary( ) function provide the coefficients and standard deviations for each model

```
> summary(ml_logit)
Call:
multinom(formula = Class ~ ., data = wine_trn)

Coefficients:
  (Intercept)  Alcohol.2 Malic.acid.       Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids. Nonflavanoid.phenols
1   -150.8796   3.719582   22.733572   63.77061          -9.551572  0.2423116     -110.51008    93.31195            -56.18379
2    198.2972 -28.033655   -1.223123 -125.48033           8.010696  1.7445375      -61.48548    69.49118            220.15011
  Proanthocyanins. Color.intensity.       Hue OD280.OD315.of.diluted.wines.   Proline.
1        -4.839092         -19.49663  38.83918                      10.19197 0.24685710
2         2.687883         -23.41452 154.80004                       2.49729 0.02028825

Std. Errors:
  (Intercept) Alcohol.2 Malic.acid.     Ash. Alcalinity.of.ash. Magnesium. Total.phenols. Flavanoids. Nonflavanoid.phenols
1    22.52277  296.3198    543.0313 40.64535           669.1547   50.97645       74.43353    156.7126             26.32834
2    11.12063  237.4662    154.1817 34.19834           286.9405  216.32121      105.05729    102.8827             38.64278
  Proanthocyanins. Color.intensity.      Hue OD280.OD315.of.diluted.wines. Proline.
1         91.26217        142.65356 13.14472                     109.24921 31.87774
2        147.80114         38.88335 18.04335                      87.27646 32.33280

Residual Deviance: 0.000008193118
AIC: 56.00001
```

R Exercise 2: Multi-class Classification

- Train the models

```
# Train multinomial logistic regression
ml_logit <- multinom(Class ~ ., data = wine_trn)

# Check the coefficients
summary(ml_logit)
t(summary(ml_logit)$coefficients)
```

✓ Coefficients of each model

```
> t(summary(ml_logit)$coefficients)
                                       1             2
(Intercept)                 -150.8796315  198.29724653
Alcohol.2                      3.7195821  -28.03365495
Malic.acid.                   22.7335724   -1.22312289
Ash.                          63.7706125 -125.48032553
Alcalinity.of.ash.            -9.5515724    8.01069633
Magnesium.                     0.2423116    1.74453745
Total.phenols.              -110.5100808  -61.48547503
Flavanoids.                   93.3119457   69.49118380
Nonflavanoid.phenols         -56.1837869  220.15010991
Proanthocyanins.              -4.8390924    2.68788272
Color.intensity.             -19.4966267  -23.41452149
Hue                           38.8391791  154.80004270
OD280.OD315.of.diluted.wines. 10.1919660    2.49729004
Proline.                       0.2468571    0.02028825
```

R Exercise 2: Multi-class Classification

- Interpret the results

```r
# Conduct 2-tailed z-test to compute the p-values
z_stats <- summary(ml_logit)$coefficients/summary(ml_logit)$standard.errors
t(z_stats)

p_value <- (1-pnorm(abs(z_stats), 0, 1))*2
options(scipen=10)
t(p_value)
```

✓ multinorm( ) does not provide the p-values, so we manually compute them

```
> t(p_value)
                                              1              2
(Intercept)                 0.00000000002098766 0.00000000000000
Alcohol.2                   0.98998474329538033 0.90602547076899
Malic.acid.                 0.96660695408866371 0.99367045293444
Ash.                        0.11665910227299237 0.00024331650010
Alcalinity.of.ash.          0.98861131348134723 0.97772785523380
Magnesium.                  0.99620734776521647 0.99356547425947
Total.phenols.              0.13762822597308633 0.55837518665469
Flavanoids.                 0.55155361898111677 0.49939557325969
Nonflavanoid.phenols        0.03284554062986977 0.00000001218935
Proanthocyanins.            0.95771272346227398 0.98549062698237
Color.intensity.            0.89129072835830669 0.54705870613885
Hue                         0.00312936175614698 0.00000000000000
OD280.OD315.of.diluted.wines. 0.92567239450692451 0.97717279806758
Proline.                    0.99382134642228603 0.99949934191516
```

R Exercise 2: Multi-class Classification

- Interpret the results

```
cbind(t(summary(ml_logit)$coefficients), t(p_value))
```

✓ Print the coefficients and p-values for each model

```
> cbind(t(summary(ml_logit)$coefficients), t(p_value))
                                     1            2                   1                2
(Intercept)                  -150.8796315  198.29724653 0.00000000002098766 0.00000000000000
Alcohol.2                       3.7195821  -28.03365495 0.98998474329538033 0.90602547076899
Malic.acid.                    22.7335724   -1.22312289 0.96660695408866371 0.99367045293444
Ash.                           63.7706125 -125.48032553 0.11665910227299237 0.00024331650010
Alcalinity.of.ash.             -9.5515724    8.01069633 0.98861131348134723 0.97772785523380
Magnesium.                      0.2423116    1.74453745 0.99620734776521647 0.99356547425947
Total.phenols.               -110.5100808  -61.48547503 0.13762822597308633 0.55837518665469
Flavanoids.                    93.3119457   69.49118380 0.55155361898111677 0.49939557325969
Nonflavanoid.phenols          -56.1837869  220.15010991 0.03284554062986977 0.00000001218935
Proanthocyanins.               -4.8390924    2.68788272 0.95771272346227398 0.98549062698237
Color.intensity.              -19.4966267  -23.41452149 0.89129072835830669 0.54705870613885
Hue                            38.8391791  154.80004270 0.00312936175614698 0.00000000000000
OD280.OD315.of.diluted.wines.  10.1919660    2.49729004 0.92567239450692451 0.97717279806758
Proline.                        0.2468571    0.02028825 0.99382134642228603 0.99949934191516
```

Coefficients (1 vs. 3)	Coefficients (2 vs. 3)	p-values (1 vs. 3)	p-values (2 vs. 3)

R Exercise 2: Multi-class Classification

- Check the classification accuracy

```
# Predict the class probability
ml_logit_haty <- predict(ml_logit, type="probs", newdata = wine_tst)
ml_logit_haty[1:10,]
```

✓ If we use type = "probs" option, the likelihood for each class is returned

```
> ml_logit_haty[1:10,]
               3 1            2
1   2.571434e-70 1  4.668832e-67
3   3.187174e-81 1  1.659844e-80
4   1.004466e-57 1 1.776978e-116
8   1.080070e-87 1  1.347261e-90
11 2.737317e-110 1  1.326322e-92
13  9.475704e-87 1  2.426455e-98
16  4.975362e-74 1  1.154407e-88
17  1.965882e-77 1  2.817637e-77
18  2.881156e-53 1  9.666261e-37
19 2.890111e-114 1 2.465335e-122
```

R Exercise 2: Multi-class Classification

- Check the classification accuracy

```
# Predict the class label
ml_logit_prey <- predict(ml_logit, newdata = wine_tst)
cfmatrix <- table(wine_tst$Class, ml_logit_prey)
cfmatrix perf_mat_wine[,2] <- perf_eval3(cfmatrix)
perf_mat_wine
```

✓ Without type = "prob" option, the class label with the highest likelihood is returned

```
> cfmatrix
 ml_logit_prey
    3  1  2
  3 12  0  0
  1  0 16  1
  2  3  0 21
> perf_eval3(cfmatrix)
[1] 0.9245283 0.9373311
```


ANY
questions?