Topic 3: Statistical Learning Models

Jonathan Ibifubara Pollyn

GitHub Repository: https://github.com/JonathanPollyn/Machine-Learning-for-Data-Science

College of Science, Engineering and Technology, Grand Canyon University

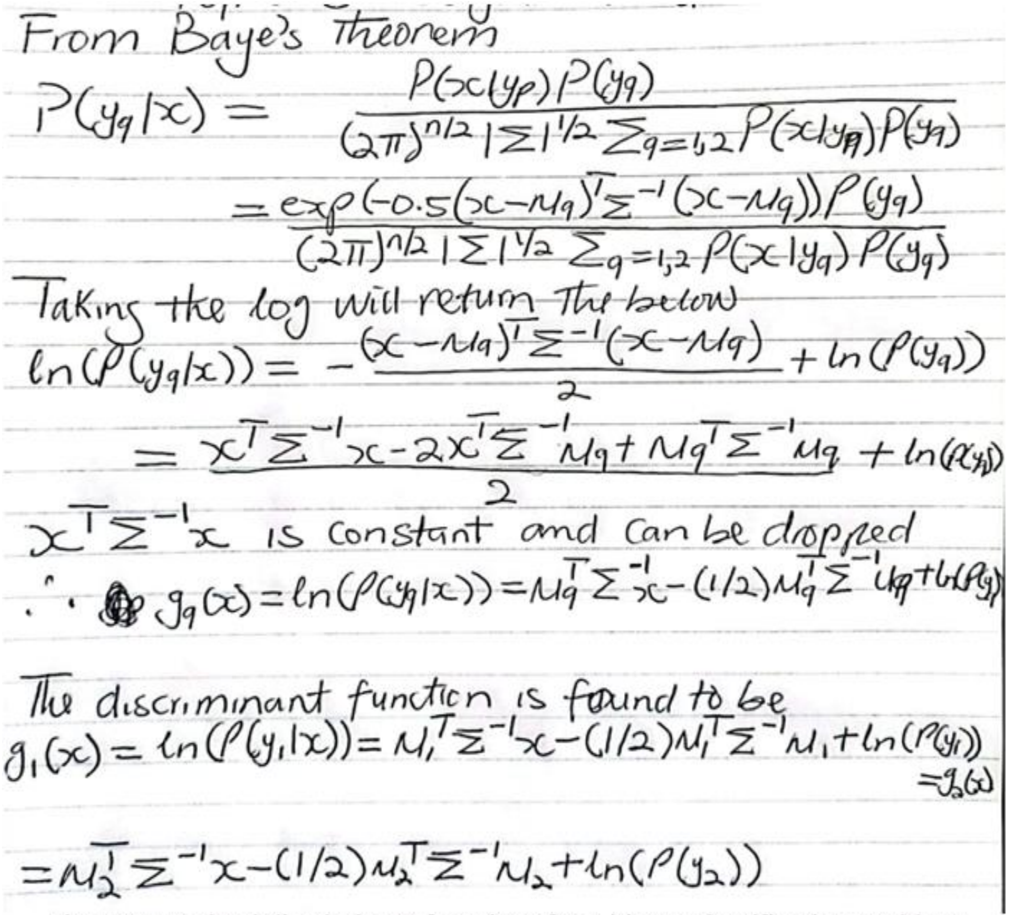
DSC-540: Machine Learning for Data Science

Dr. Aiman Darwiche

November 17, 2021

Discriminant functions for a binary classification

Linear discriminant analysis is a technique for classifying, reducing dimensions, and visualizing data. It has been in existence for quite some time. Despite its simplicity, LDA frequently yields reliable, reasonable, and understandable classification results. The discriminant function determines how probable data x is to belong to each of the classes. The set of x where two discriminant functions have the same value is the decision boundary dividing any two classes, k and l. As a result, any data falling on the decision border is equally likely to come from one of the two groups. LDA arises in the case where we assume equal covariance among K classes. That is, instead of one covariance matrix per class, all classes have the same covariance matrix. The decision boundary between any pair of classes is also a linear function in x. Without the equal covariance assumption, the quadratic term in the likelihood does not cancel out; hence the resulting discriminant function is quadratic in x. Figure 1 shows the derived discriminant functions for a binary classification problem and after the simplification we get the desired decision boundary.



From Baye's theorem

$$P(y_q|x) = \frac{P(x|y_p)P(y_q)}{(2\pi)^{n/2}|\Sigma|^{1/2}\sum_{q=1,2}P(x|y_q)P(y_q)}$$

$$= \frac{\exp(-0.5(x-\mu_q)^T\Sigma^{-1}(x-\mu_q))P(y_q)}{(2\pi)^{n/2}|\Sigma|^{1/2}\sum_{q=1,2}P(x|y_q)P(y_q)}$$

Taking the log will return the below

$$\ln(P(y_q|x)) = -\frac{(x-\mu_q)^T\Sigma^{-1}(x-\mu_q)}{2} + \ln(P(y_q))$$

$$= \frac{x^T\Sigma^{-1}x - 2x^T\Sigma^{-1}\mu_q + \mu_q^T\Sigma^{-1}\mu_q}{2} + \ln(P(y_q))$$

$x^T\Sigma^{-1}x$ is constant and can be dropped

$$\therefore g_q(x) = \ln(P(y_q|x)) = \mu_q^T\Sigma^{-1}x - (1/2)\mu_q^T\Sigma^{-1}\mu_q + \ln(P(y_q))$$

The discriminant function is found to be

$$g_1(x) = \ln(P(y_1|x)) = \mu_1^T\Sigma^{-1}x - (1/2)\mu_1^T\Sigma^{-1}\mu_1 + \ln(P(y_1))$$

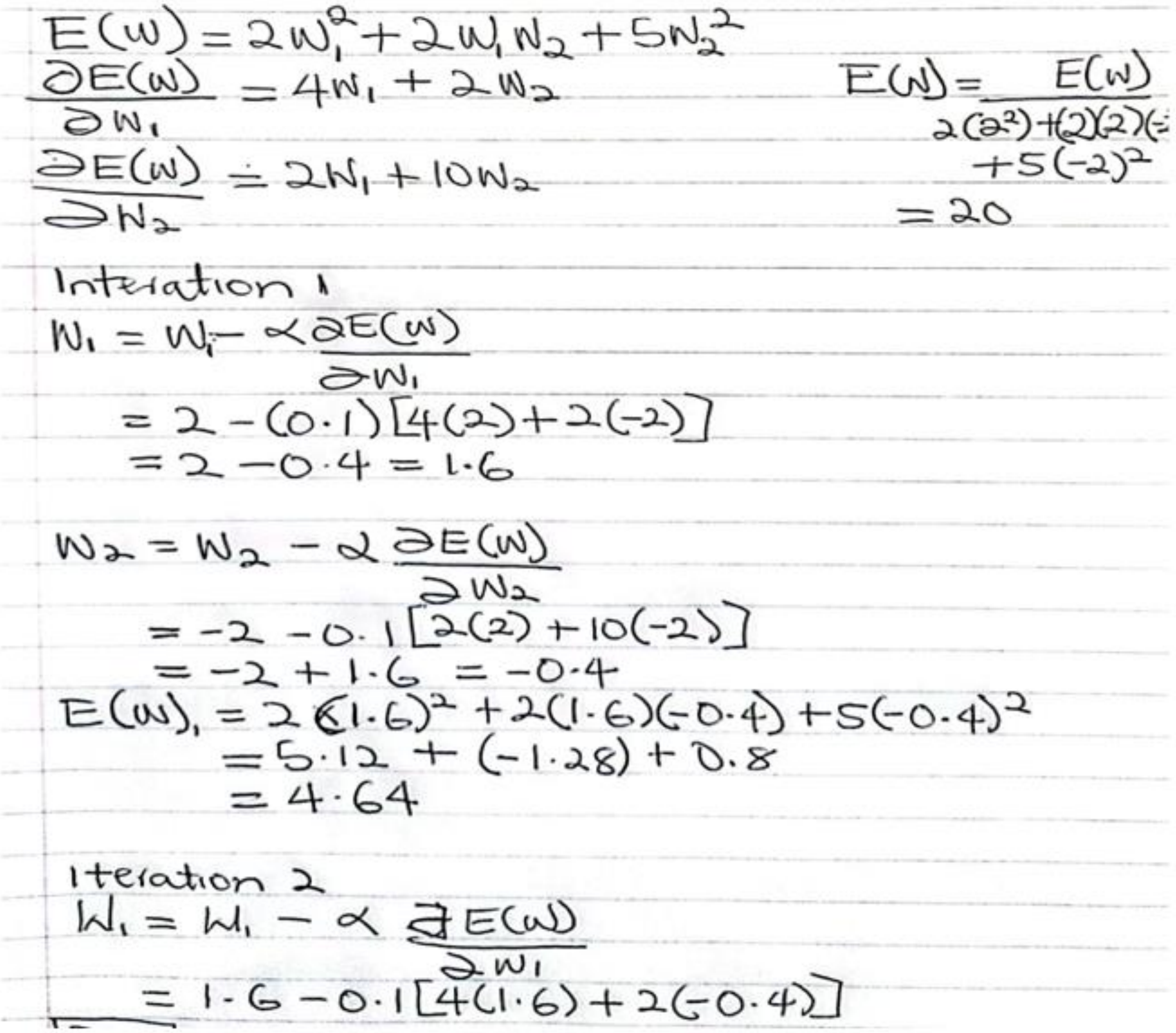
$$= g_2(x)$$

$$= \mu_2^T\Sigma^{-1}x - (1/2)\mu_2^T\Sigma^{-1}\mu_2 + \ln(P(y_2))$$

Figure 1: Derived discriminant functions for a binary classification problem

Two iterations of the gradient algorithm to find the minima

In machine learning and deep learning, gradient descent is the most used optimization algorithm. It's a first-order optimization procedure. This implies that while updating the parameters, it only considers the first iteration. We update the parameters in the opposite direction of the gradient of the objective function. The parameters on each iteration, where the gradient indicates the sharpest ascending direction. The learning rate determines the number of steps we take each iteration to attain the local minimum. As a result, we descend in the direction of the slope until we hit a local minimum. Figures 2.1 and 2.2 show two iterations of the gradient, first setting the weight w and the bias b to any random values. Decide on a learning rate value. The learning rate controls the size of each iteration's step. If α is very tiny, convergence will take a long time and will be computationally costly. If α is big, it is possible that it will fail to converge and will exceed the minimum.



$E(w) = 2w_1^2 + 2w_1 w_2 + 5w_2^2$

$\frac{\partial E(w)}{\partial w_1} = 4w_1 + 2w_2$

$\frac{\partial E(w)}{\partial w_2} = 2w_1 + 10w_2$

$E(w) = \underline{\quad E(w)}$

$2(2^2) + (2)(2)(-$

$+5(-2)^2$

$= 20$

Interation 1

$w_1 = w_i - \alpha \frac{\partial E(w)}{\partial w_1}$

$= 2 - (0.1)[4(2) + 2(-2)]$

$= 2 - 0.4 = 1.6$

$w_2 = w_2 - \alpha \frac{\partial E(w)}{\partial w_2}$

$= -2 - 0.1[2(2) + 10(-2)]$

$= -2 + 1.6 = -0.4$

$E(w)_1 = 2(1.6)^2 + 2(1.6)(-0.4) + 5(-0.4)^2$

$= 5.12 + (-1.28) + 0.8$

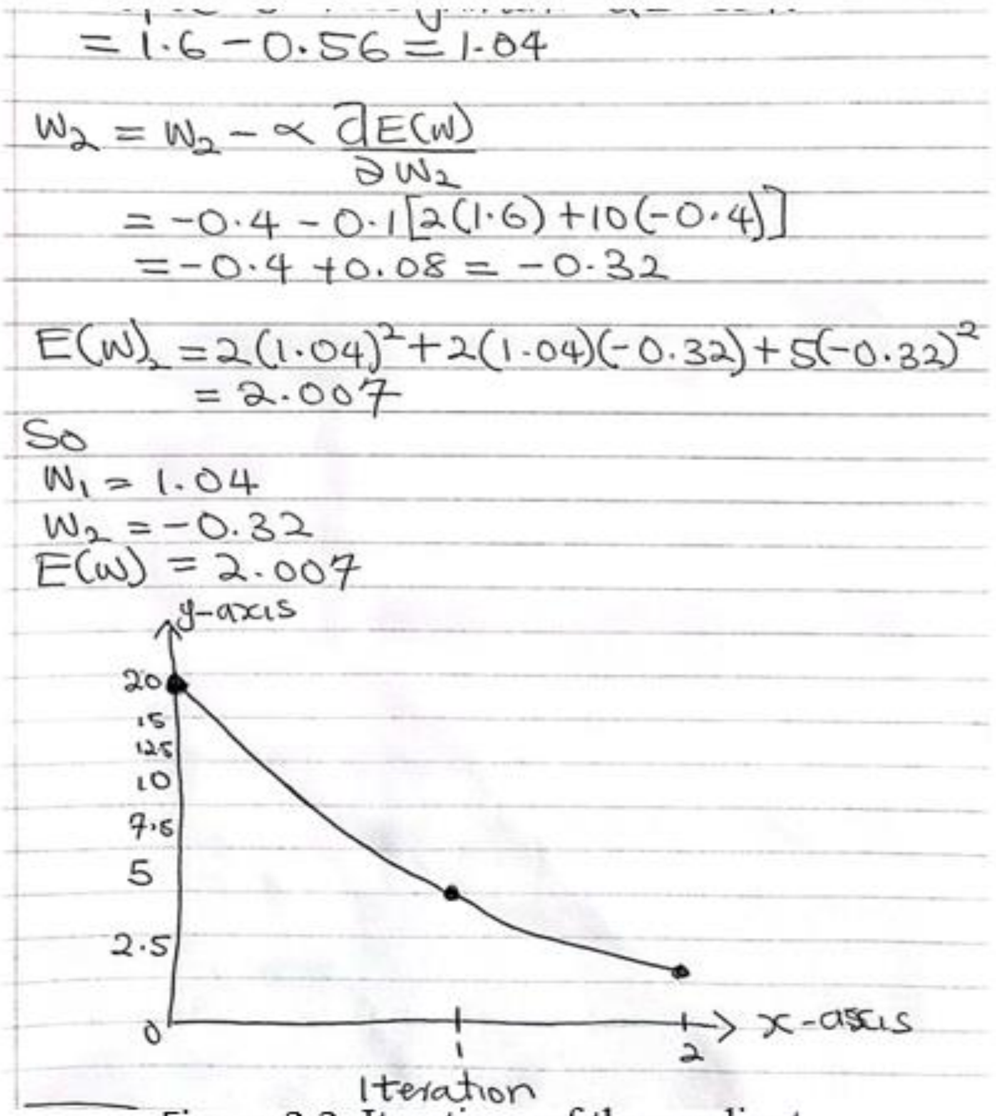
$= 4.64$

Iteration 2

$w_1 = w_1 - \alpha \frac{\partial E(w)}{\partial w_1}$

$= 1.6 - 0.1[4(1.6) + 2(-0.4)]$

Figure 2.1: Iterations of the gradient



$= 1.6 - 0.56 = 1.04$

$w_2 = w_2 - \alpha \frac{\partial E(w)}{\partial w_2}$

$= -0.4 - 0.1[2(1.6) + 10(-0.4)]$

$= -0.4 + 0.08 = -0.32$

$E(w)_2 = 2(1.04)^2 + 2(1.04)(-0.32) + 5(-0.32)^2$

$= 2.007$

So

$w_1 = 1.04$

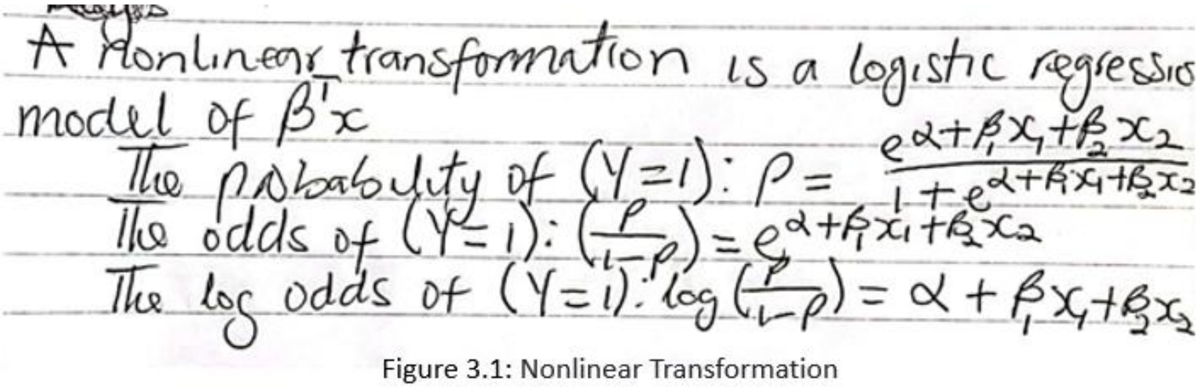
$w_2 = -0.32$

$E(w) = 2.007$

Figure 2.2: Iterations of the gradient

Logistic regression as a nonlinear regression problem

Traditionally, logistic regression has been used to create a hyperplane that divides the feature space into classes. If we think that the decision boundary is nonlinear, we can try different nonlinear functional forms for the logit function to achieve better results. In terms of odds and probability, logistic regression is nonlinear, but it is linear in terms of log odds. Refer to figure 3.1 for the nonlinear transformation of logistic regression. Figure 3.2 shows a probability plot P(Y = 1) as a function of X that indicates a nonlinear relationship. Also, figure 3.3 shows the probability of Y being 1 given X is proving nonlinear as well. Finally, there is a linear connection between the log odds of Y being 1 and the log odds of Y being 1.



A Nonlinear transformation is a logistic regressio
model of $\beta^T x$

The probability of $(Y=1)$: $P = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2}}$

The odds of $(Y=1)$: $\left(\frac{P}{1-P}\right) = e^{\alpha+\beta_1 x_1+\beta_2 x_2}$

The log odds of $(Y=1)$: $\log\left(\frac{P}{1-P}\right) = \alpha+\beta_1 x_1+\beta_2 x_2$

Figure 3.1: Nonlinear Transformation

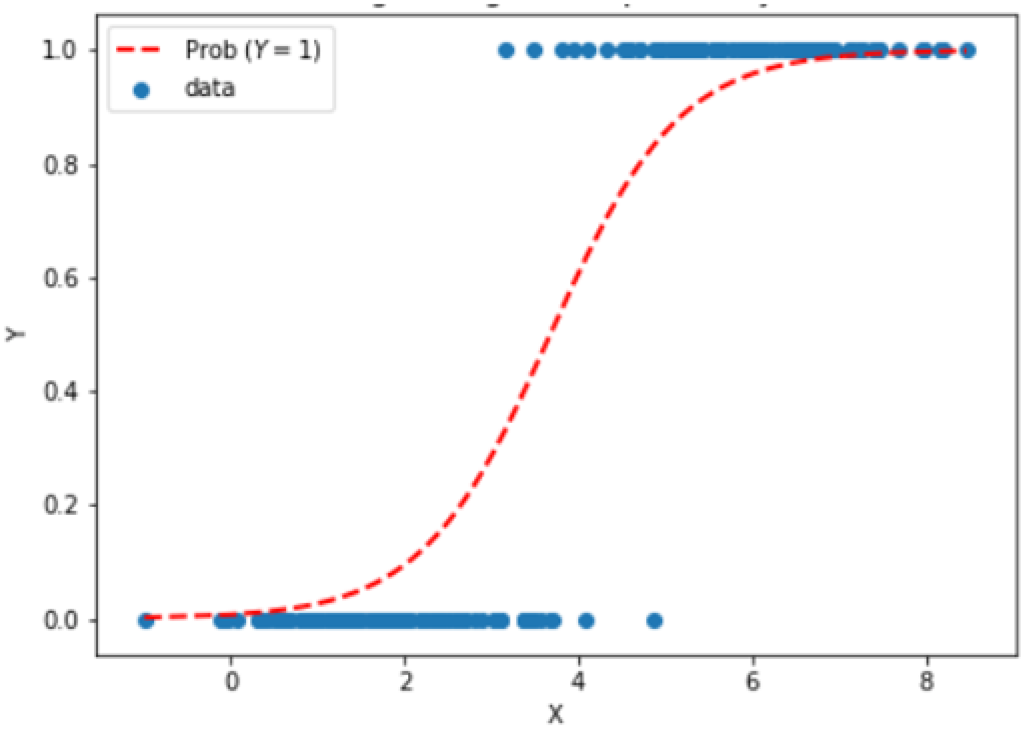


Figure 3.2: Nonlinear relationship

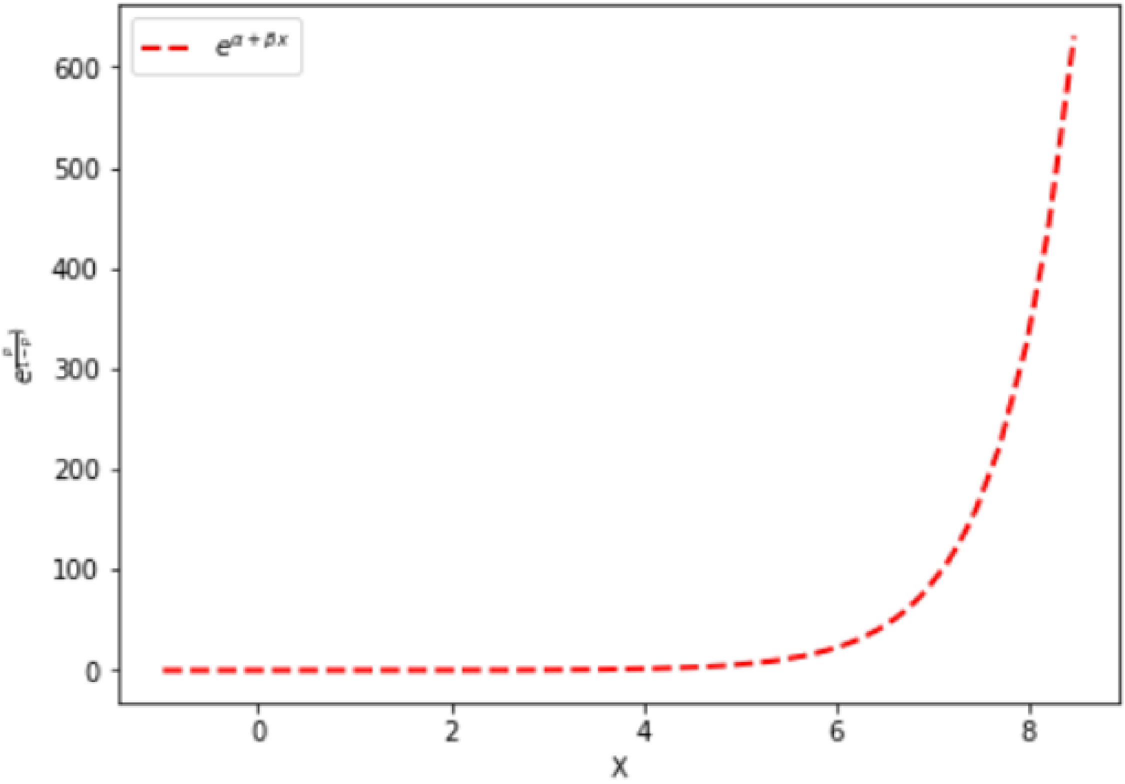


Figure 3.3: Nonlinear relationship

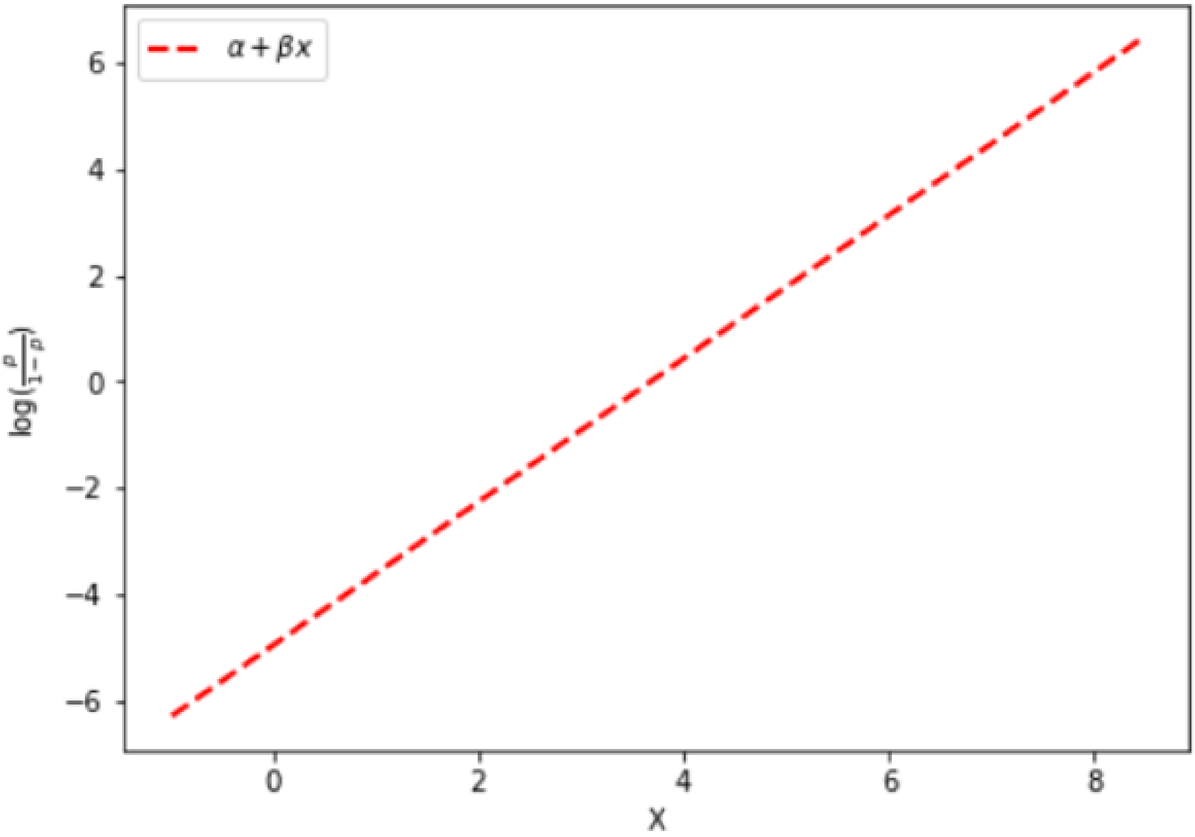


Figure 3.4: linear relationship

References List

Dabbura, I. (2017, December 21). *Gradient descent algorithm and its variants | by imad dabbura | towards data science*. Medium. https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3

Self study - logistic regression is a nonlinear regression problem? - cross validated. (n.d.). Cross Validated. https://stats.stackexchange.com/questions/365391/logistic-regression-is-a-nonlinear-regression-problem

Xiaozhou, Y. (2020, May 9). *Linear discriminant analysis, explained | by yang xiaozhou | towards data science*. Medium. https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b