Credit card fraud detection using Logistic Regression model

Dr. Pritam Anand.
Assistant Professor,
DA-IICT, Gandhinagar.

Lottery Fraud

Juice Jacking

Card Skimming

Credit Card Fraud

Phishing Link

Unknown mobile apps

Vishing Calls

Federal Trade Commission (FTC) USA study [1]

- Nearly half of all American adults have had a fraudulent charge on their cards, amounting to around 127 million people. More than one in three card holders has experienced card fraud more than once[2].

- The median fraudulent credit card charge was $62 which approximate $8 billion in fraudulent charges among all American consumers[2].

- The credit card fraud has also become even more common since the start of the pandemic.

- Reports of credit card fraud increased by 44 percent between 2019 and 2020 according to the Federal Trade Commission (FTC) [1] .

1. https://www.ftc.gov/system/files/documents/reports/consumer-sentinel-network-data-book-2020/csn_annual_data_book_2020.pdf
2. https://www.security.org/digital-safety/credit-card-fraud-report/#references.

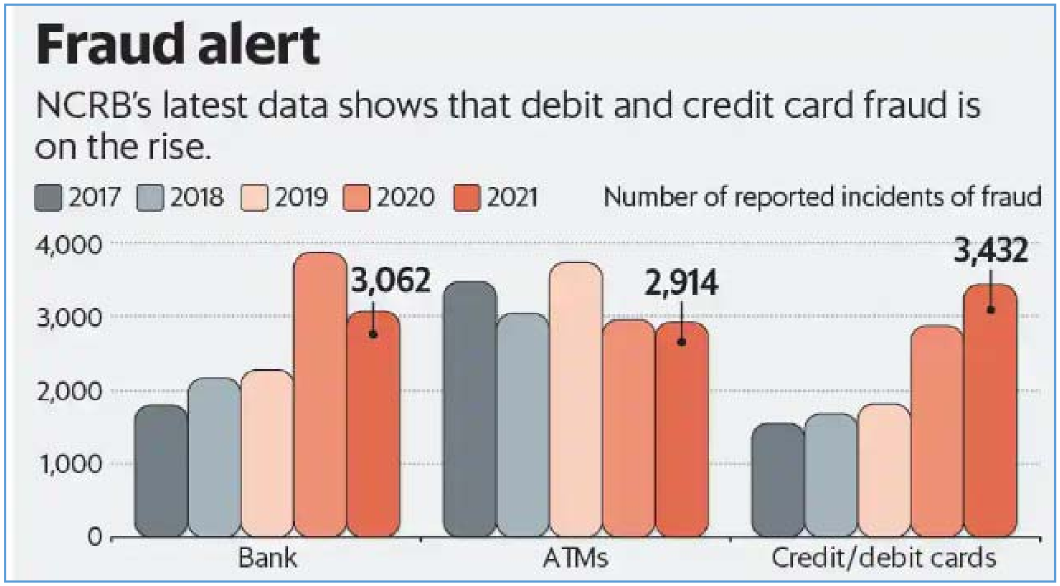
Scammers are growing smart..

They’ll take notice of any new trends, like stimulus checks, unemployment payments, and general consumer financial trends, and capitalize upon them with phishing scams, identity theft, or other types of cybercrime.

Hari Ravichandran

CEO, AURA
Identity Theft and fraud protection company

Indian Picture..

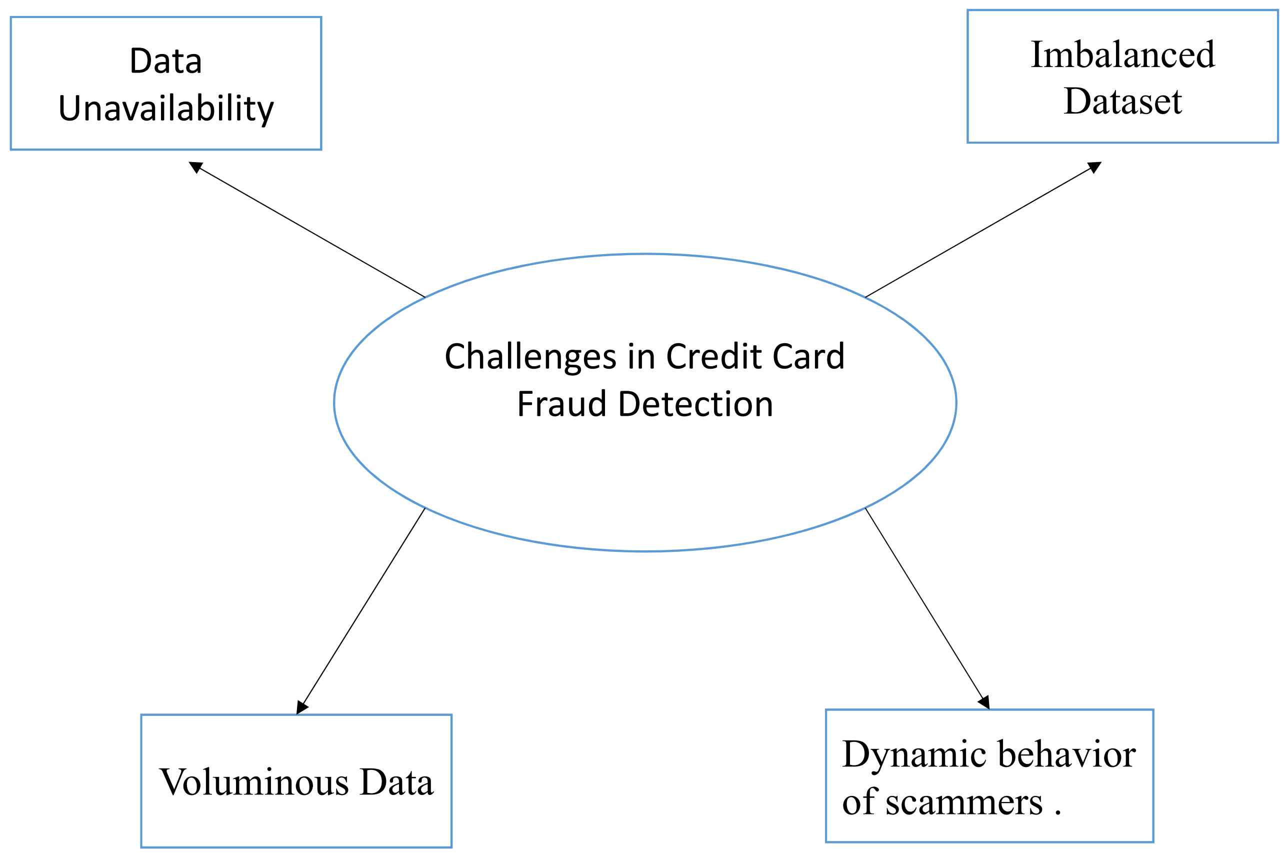


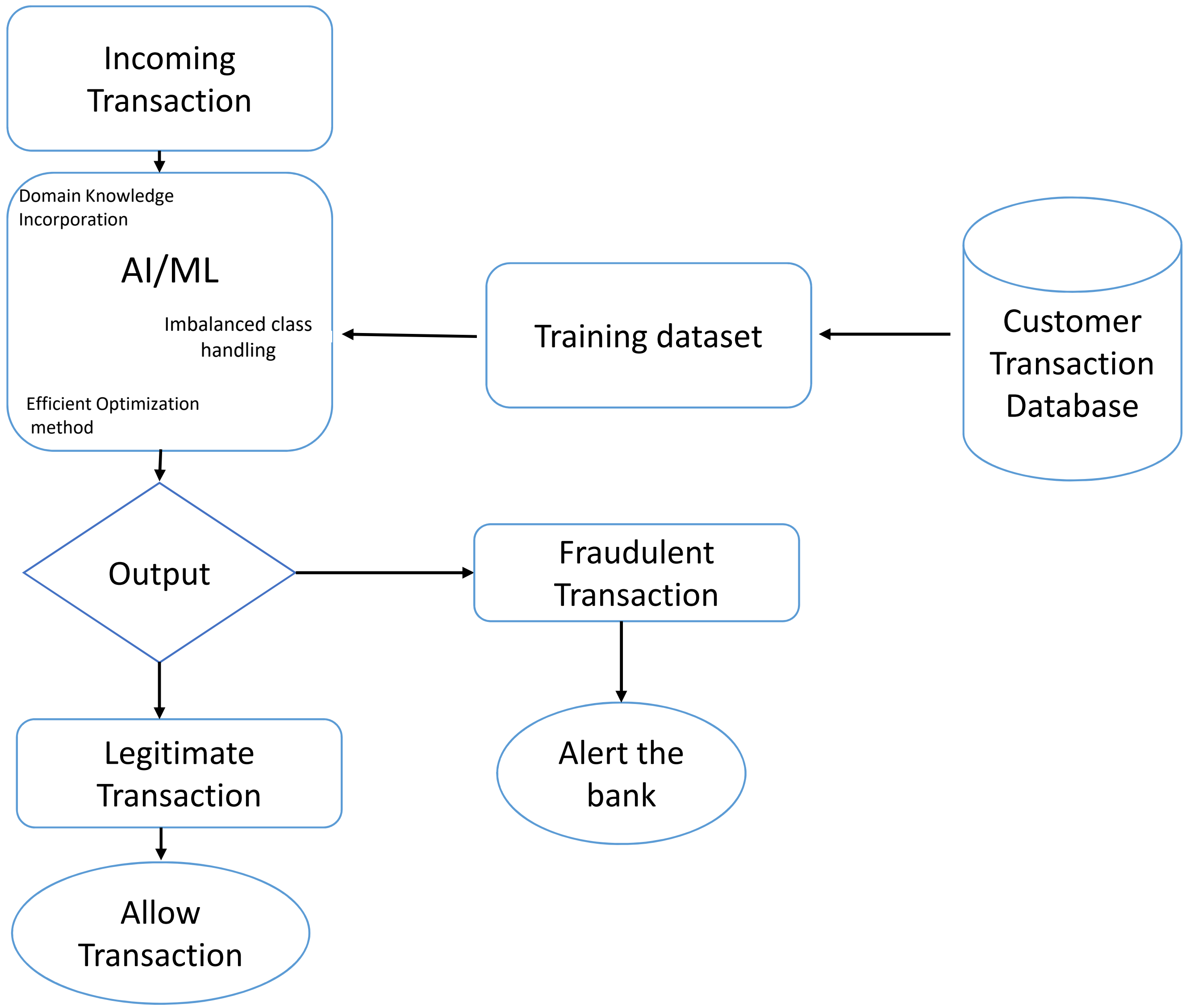
Report of National Crime Records Bureau, India [3]

According to the data, 3,432 cases of credit and debit card frauds were filed from across India in 2021, up nearly 20% from the year-earlier. In 2020, such frauds increased by over 70% [3] *.*

[3] https://www.livemint.com/industry/banking/debit-credit-card-frauds-on-rise-atm-scams-down-ncrb-11661885877307.html

Challenges in Credit Card Fraud Detection

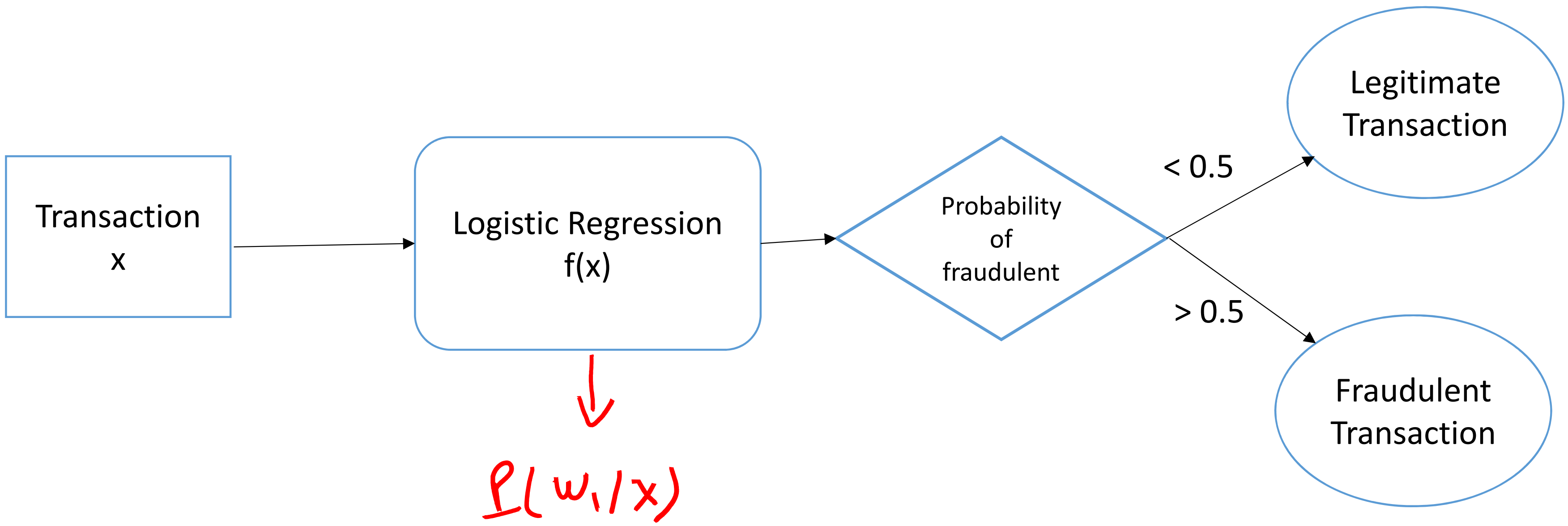


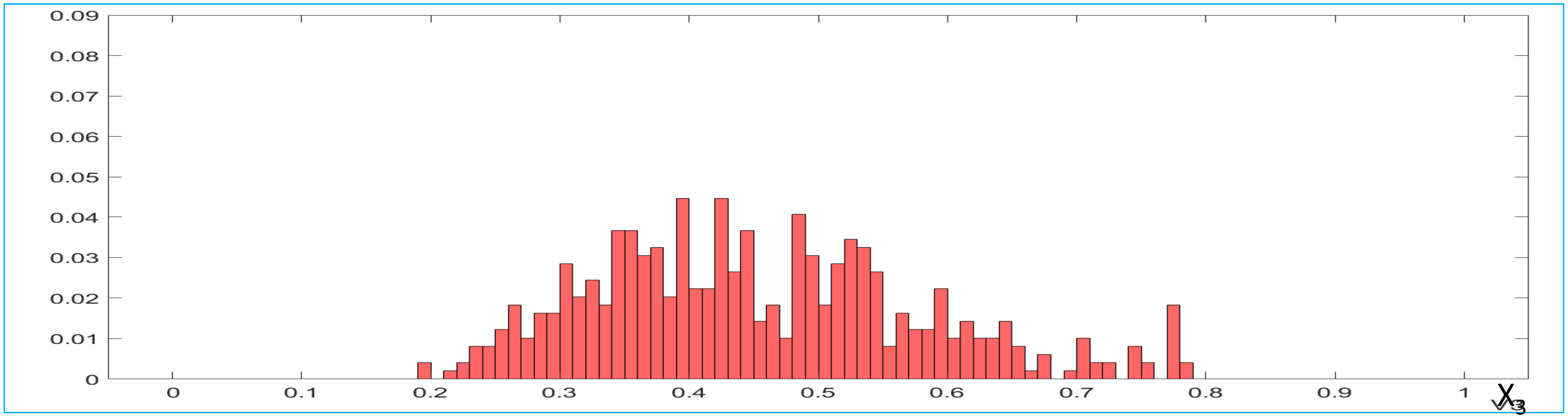

Incoming Transaction
Domain Knowledge Incorporation
AI/ML
Imbalanced class handling
Efficient Optimization method
Training dataset
Customer Transaction Database
Output
Fraudulent Transaction
Alert the bank
Legitimate Transaction
Allow Transaction

Real World Dataset

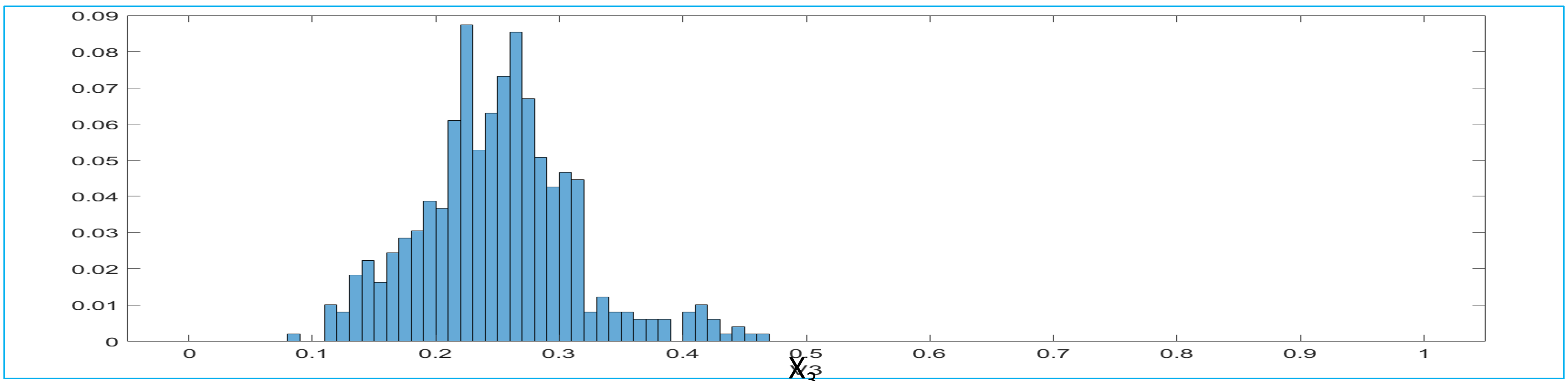
- The dataset is publicly available at https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?select=creditcard.csv
- It contains transactions made by credit cards in September 2013 by European cardholder in two days.
- Due to privacy issue, it does not provide original variables which provides the background information about the transaction.
- It contains 30 variables ,out of them $X_1,X_2 ,\ldots, X_{28}$ are transformed variables with Principal Component Analysis.
- Only Time and Amount is original variable which provides real information about the transaction.
- The dataset contains 492 fraudulent transaction out of 284807 transactions. The positive class account for 0.172% of all transaction.

Logistic Regression for credit card fraud detection

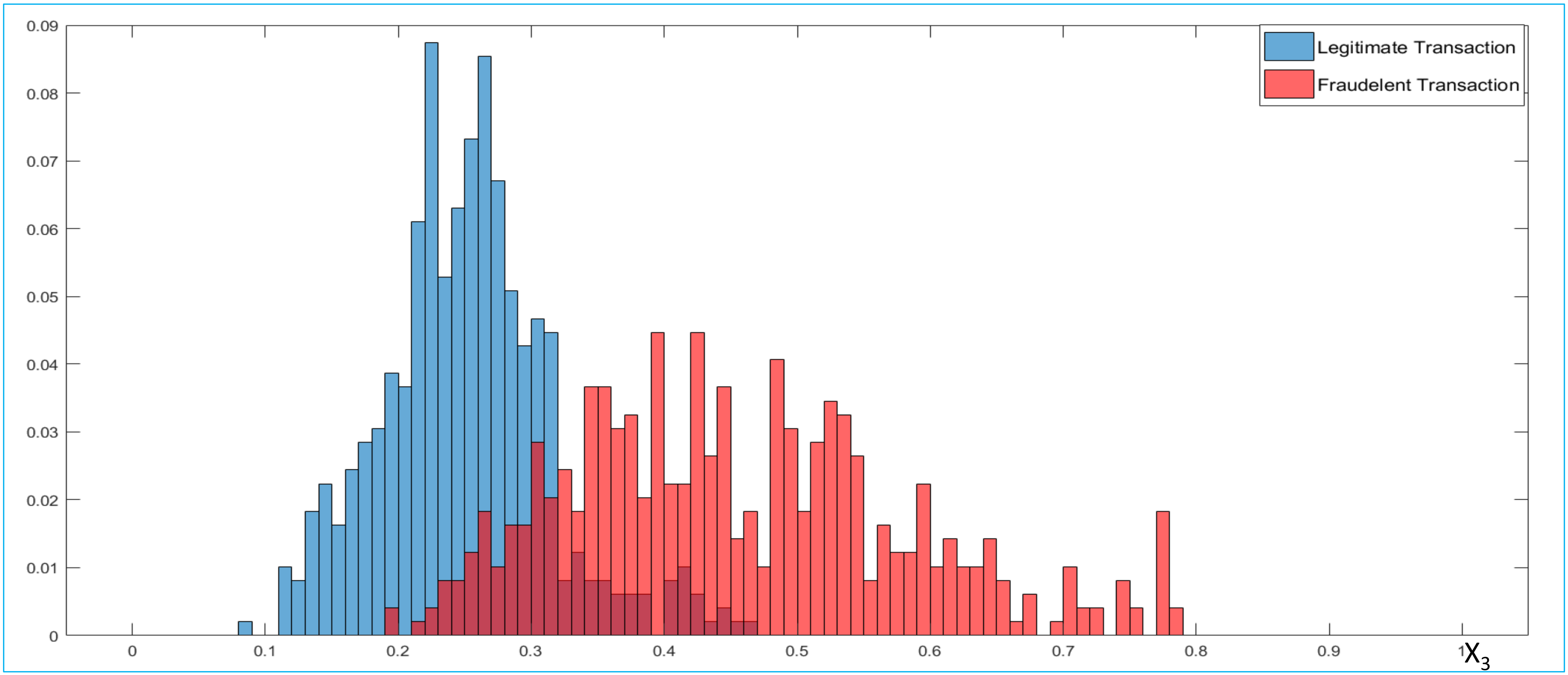


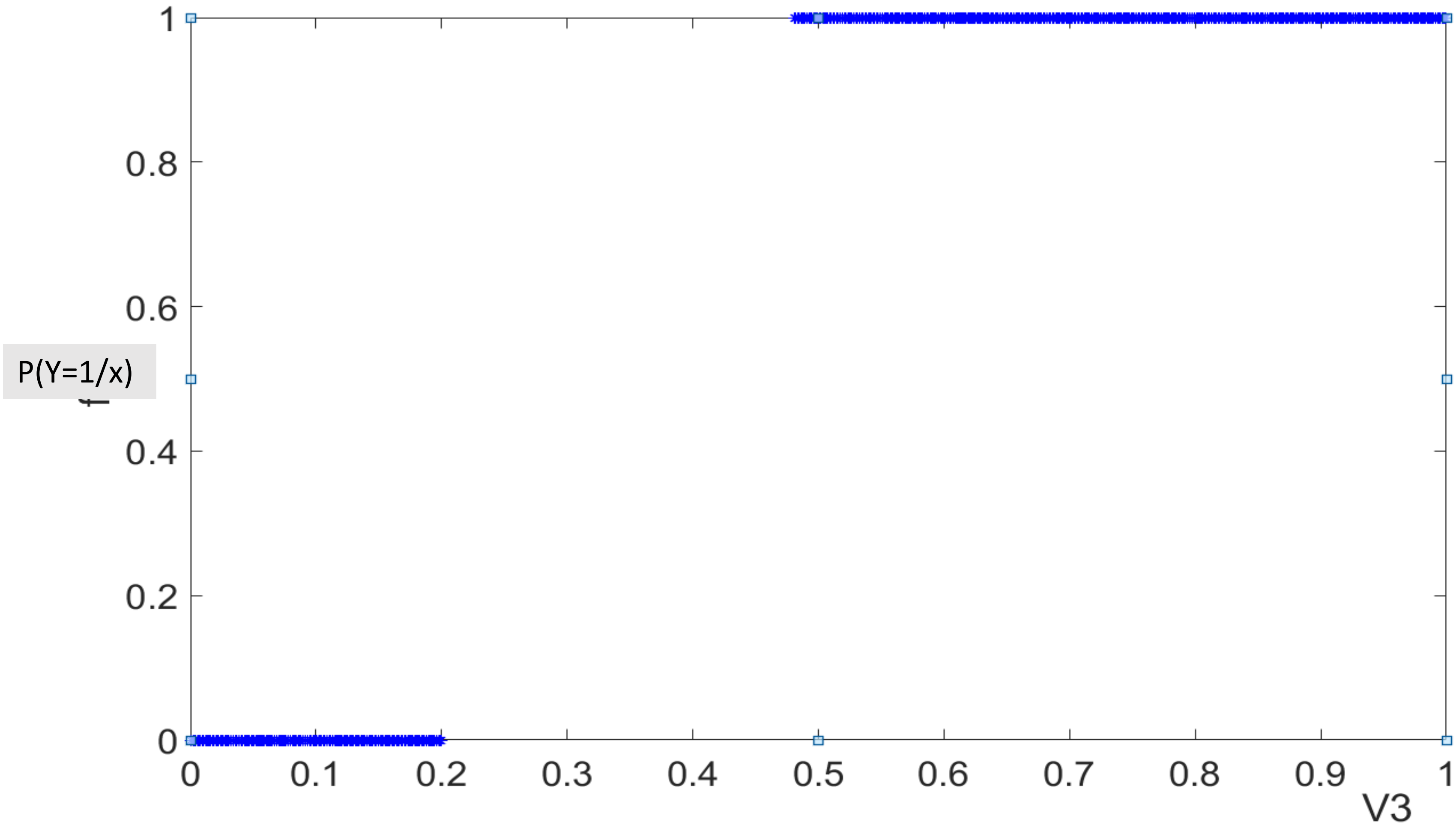


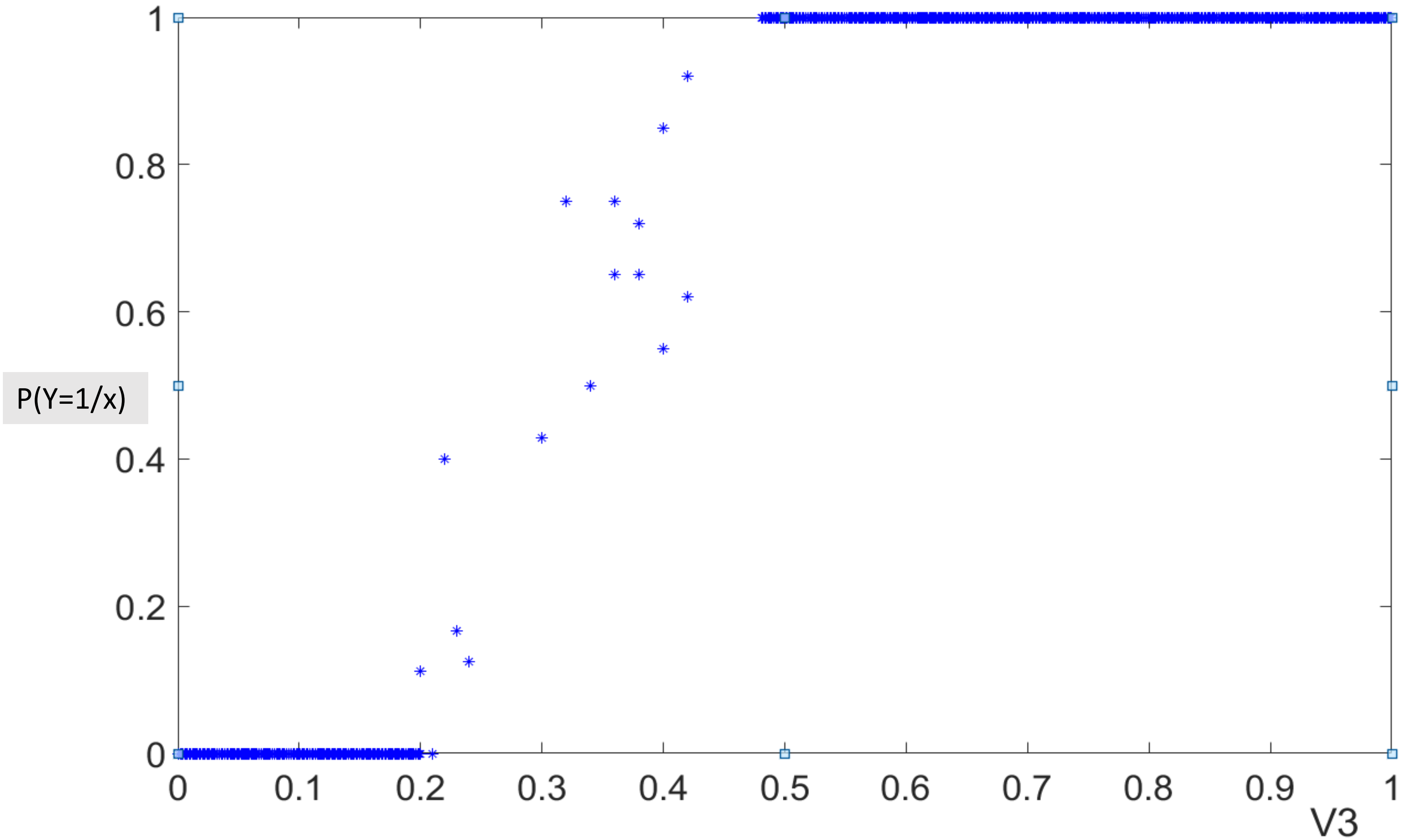
Fraudulent Transactions



Legitimate Transactions


Legitimate Transaction
Fraudelent Transaction
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X_3


P(Y=1/x)
V3
0
0.2
0.4
0.6
0.8
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



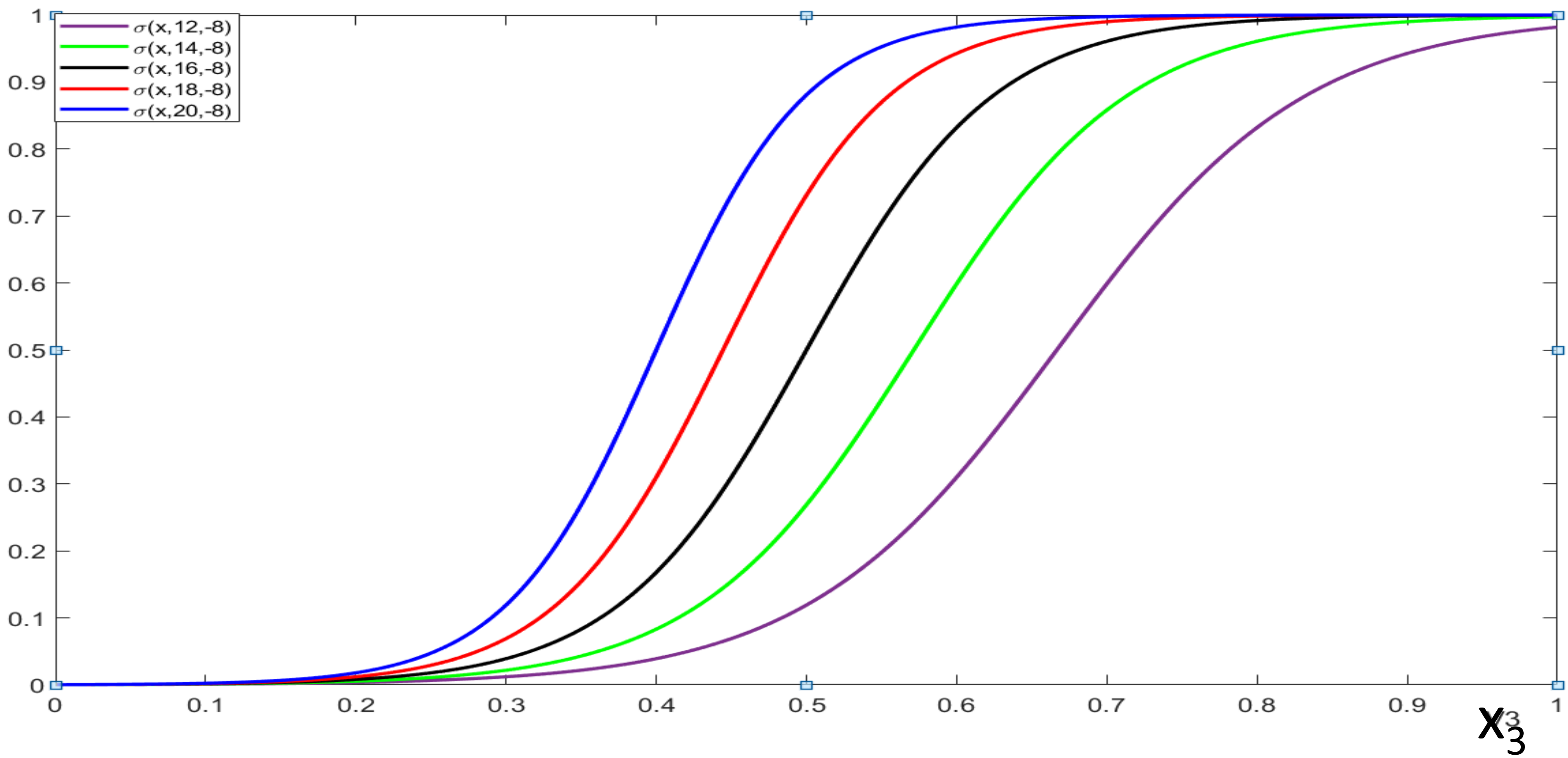
P(Y=1/x)
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
V3

Sigmoidal Function $\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{1+e^{x}}$

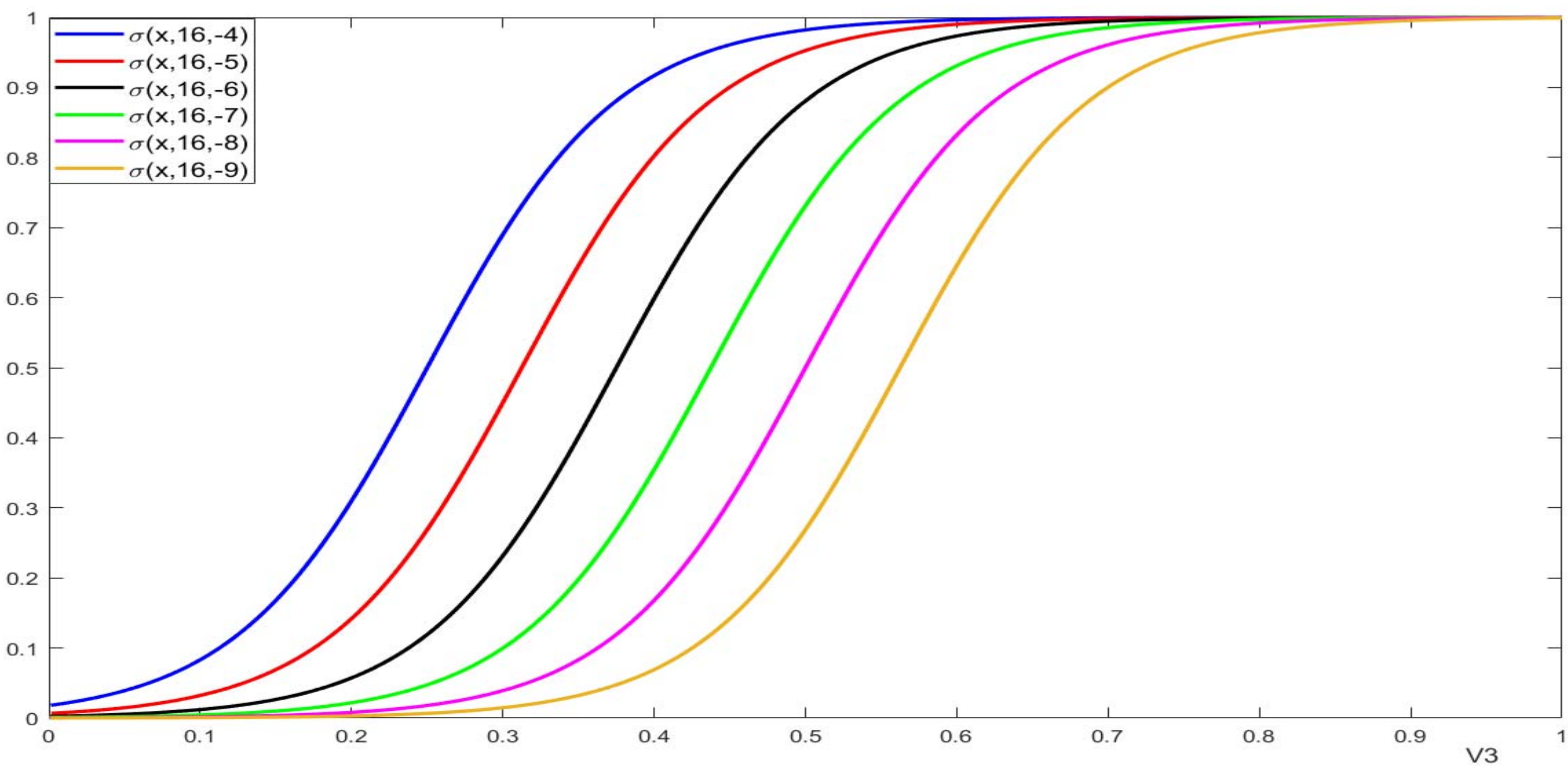
$\sigma'(x) = \sigma(x)\,(1-\sigma(x))$

$$\frac{1}{1+e^{-(\beta_1 x+\beta_0)}} = \frac{e^{(\beta_1 x+\beta_0)}}{1+e^{(\beta_1 x+\beta_0)}}$$

https://www.desmos.com/calculator/coknirwubg

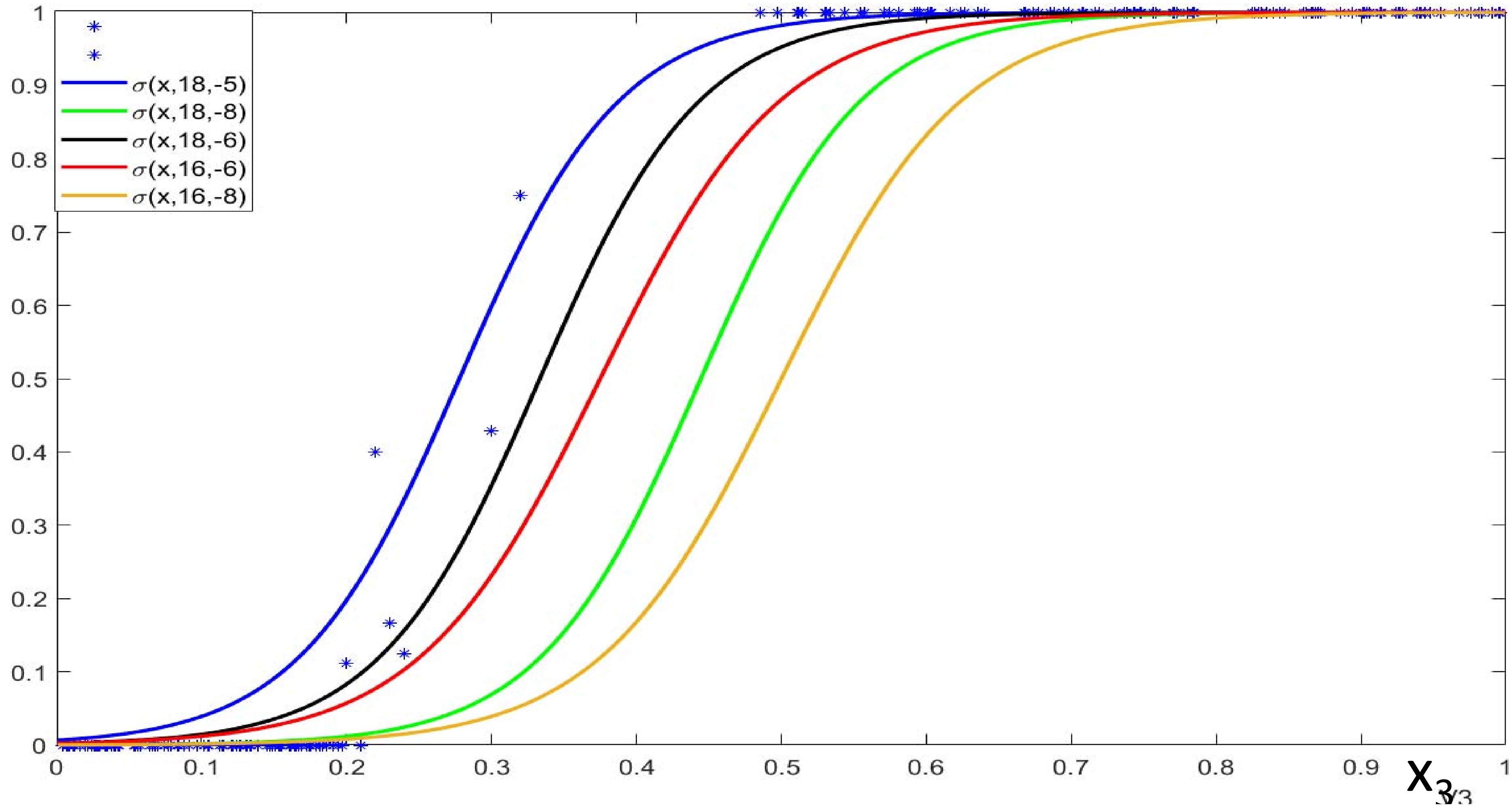


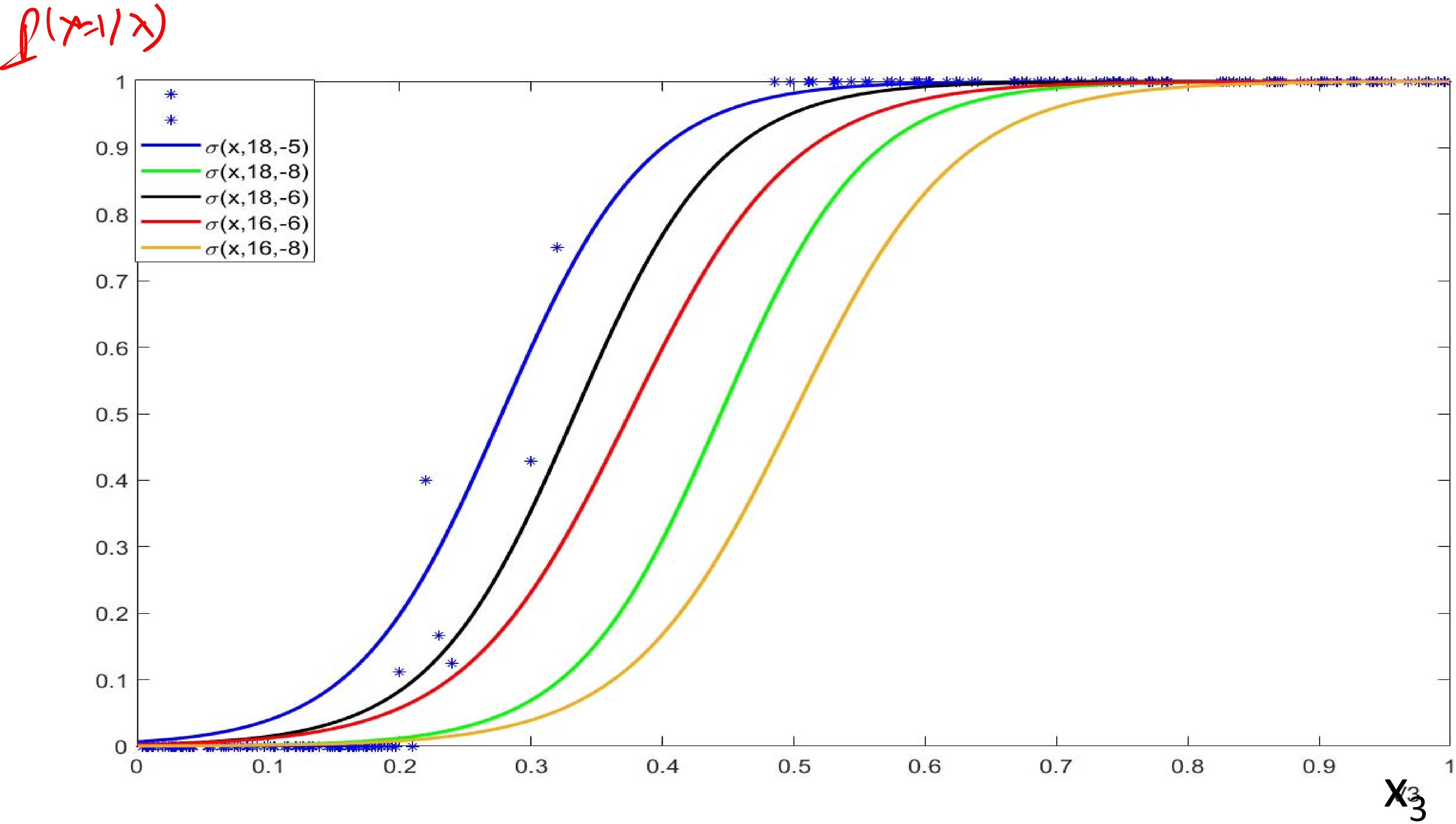
σ(x,12,-8)
σ(x,14,-8)
σ(x,16,-8)
σ(x,18,-8)
σ(x,20,-8)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x3



σ(x,16,-4)
σ(x,16,-5)
σ(x,16,-6)
σ(x,16,-7)
σ(x,16,-8)
σ(x,16,-9)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
V3

Logistic Function fitting





p(y=1|x)
σ(x,18,-5)
σ(x,18,-8)
σ(x,18,-6)
σ(x,16,-6)
σ(x,16,-8)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x3

Logistic Regression on Test data

$$f(x) = \frac{1}{1+e^{-(\beta_1 x + \beta_0)}}$$

Estimated

$\beta_1 = 18$

$\beta_0 = -5$

Test Transaction $x_3 = 0.2$ →

Logistic Regression on Test data

Estimated

$\beta_1 = 18$

$\beta_0 = -5$

Test Transaction

$x_3 = 0.2$

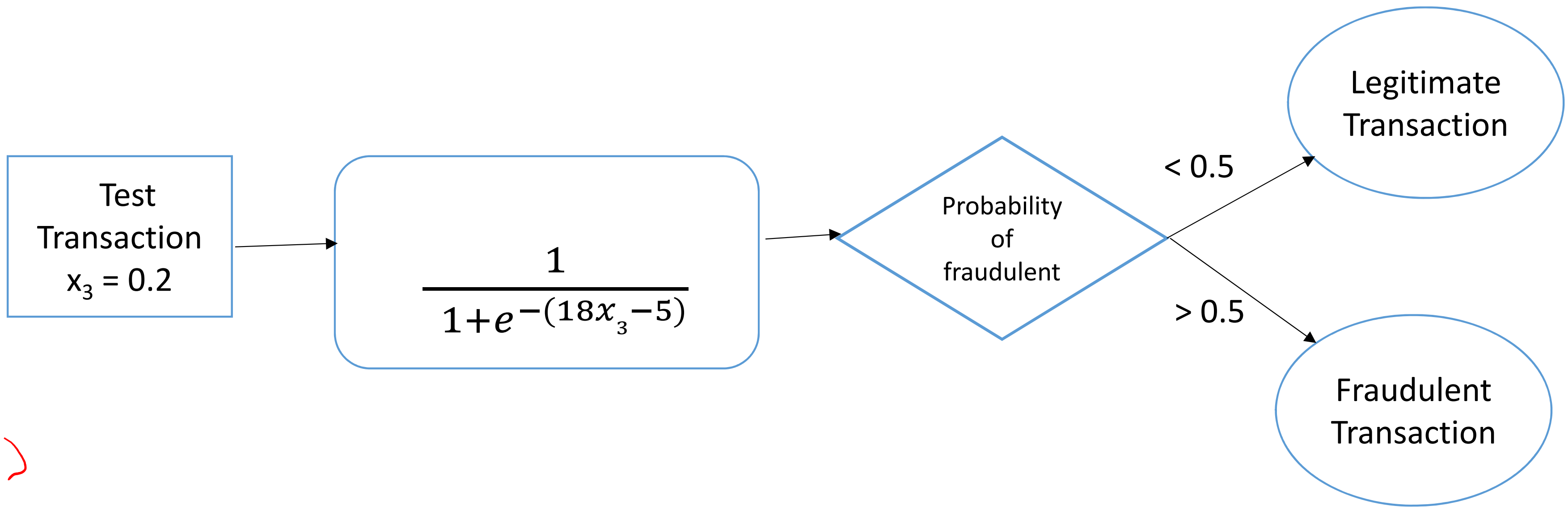
$$f(x_3) = \frac{1}{1+e^{-(18x_3-5)}}$$

Logistic Regression on Test data

Estimated

$\beta_1 = 18$

$\beta_0 = -5$

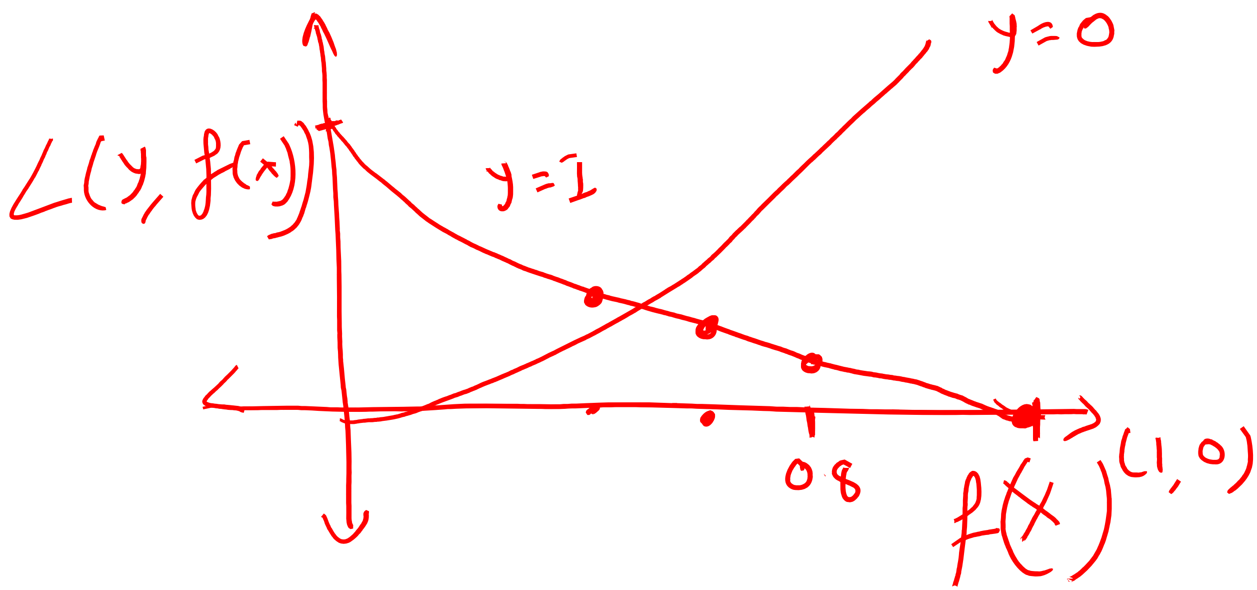


Fraudulent Transaction

Logistic Regression and Cross entropy Loss

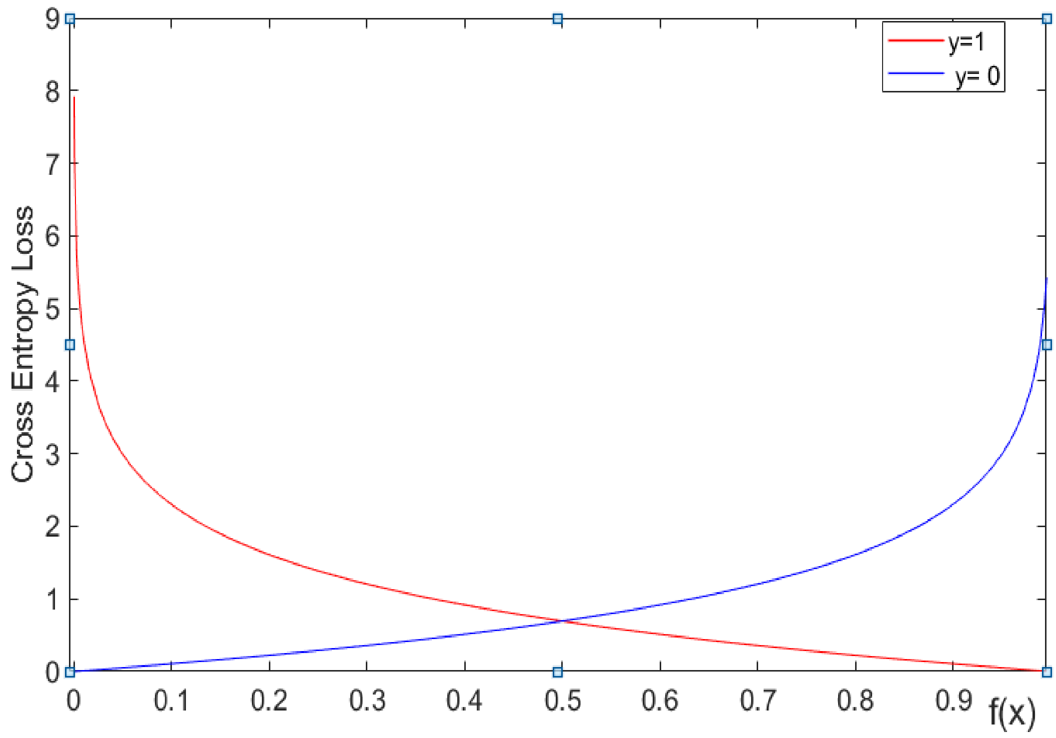
$$L\big(y, f(x)\big) = \begin{cases} -\log\big(f(x)\big) & if \quad y = 1 \\ -\log\big(1 - f(x)\big) & if \quad y = 0 \end{cases}$$

$f(x) = \underline{P(y=1/x)}$



Logistic Regression and Cross entropy Loss

$$L\big(y, f(x)\big) = \begin{cases} -\log\big(f(x)\big) & if \quad y = 1 \\ -\log\big(1 - f(x)\big) & if \quad y = 0 \end{cases}$$



Logistic Regression and Cross entropy Loss

$$L(y, f(x)) = \begin{cases} -\log(f(x)) & if \quad y = 1 \\ -\log(1 - f(x)) & if \quad y = 0 \end{cases}$$

$$= -ylog(f(x)) - (1-y)\log(1 - f(x))$$

$$P(y=1/x) = \frac{1}{1 + e^{-(\beta_1 x + \beta_0)}}$$

Logistic Regression and Cross entropy Loss

$$L\big(y, f(x)\big) = \begin{cases} -\log\big(f(x)\big) & if \quad y = 1 \\ -\log\big(1 - f(x)\big) & if \quad y = 0 \end{cases}$$

$$= -ylog\big(f(x)\big) \; - \; (1 - y)\log(1 - f(x))$$

$$\underset{f}{Min} \quad \sum_{i=1}^{l} -y_i log\big(f(x_i)\big) \; - \; (1 - yi)\log(1 - f(x_i))$$

Logistic Regression and Cross entropy Loss

$f(x) = \frac{1}{1+e^{-(\beta_1 x + \beta_0)}} = \frac{e^{\beta_1 x + \beta_0}}{1+e^{\beta_1 x + \beta_0}}$

$$\underset{f}{Min} \quad \sum_{i=1}^{l} -y_i log\big(f(x_i)\big) - (1 - yi)\log(1 - f(x_i))$$

$$\underset{(\beta_1, \beta_0)}{Min} \quad \sum_{i=1}^{l} -y_i log\left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0)}}\right) - (1 - yi) log\left(1 - \frac{1}{1+e^{-(\beta_1 x_i + \beta_0)}}\right)$$

$$\sum_{i=1}^{l}\left[\ln\left(1+e^{\beta_0+\beta_1 x_i}\right) - y_i(\beta_0+\beta_1 x_i)\right]$$

$$\rightarrow \sum_{i=1}^{l}\left[-y_i \ln\left(\frac{e^{z_i}}{1+e^{z_i}}\right) - (1-y_i)\ln\left(\frac{1}{1+e^{z_i}}\right)\right]$$

$$(z_i = \beta_0+\beta_1 x_i)$$

$$= \sum_{i=1}^{l}\Big[-y_i\ln(e^{z_i}) + y_i \cancel{\ln(1+e^{z_i})}$$

$$+ \ln(1+e^{z_i}) - y_i \cancel{\ln(1+e^{z_i})}\Big]$$

$$= \sum_{i=1}^{L} \left(\ln\left(1 + e^{\beta_0 + \beta_1 x_i}\right) - y_i(\beta_0 + \beta_1 x_i) \right)$$

Logistic Regression and Cross entropy Loss

$$\frac{1}{1+e^{-(\beta_1 x+\beta_0)}} = \frac{e^{\beta_1 x+\beta_0}}{1+e^{\beta_1 x+\beta_0}}$$

$$\underset{f}{Min} \quad \sum_{i=1}^{l} -y_i log(f(x_i)) - (1-yi)\log(1-f(x_i))$$

$$\underset{(\beta_1,\beta_0)}{Min} \sum_{i=1}^{l} -y_i log\left(\frac{1}{1+e^{-(\beta_1 x_i+\beta_0)}}\right) - (1-yi)log\left(1-\frac{1}{1+e^{-(\beta_1 x_i+\beta_0)}}\right)$$

$$= \underset{(\beta_1,\beta_0)}{Min} \; -\sum_{i=1}^{l}\left(y_{i(}\beta_1 x_i+\beta_{0)} - log\left(1+e^{(\beta_1 x_i+\beta_0)}\right)\right)$$

✓

$$-y_i \log\left(\frac{e^{\beta_1 x+\beta_0}}{1+e^{\beta_1 x+\beta_0}}\right) - (1-y_i)\log\left(\frac{e^{-(\beta_1 x+\beta_0)}}{1+e^{-(\beta_1 x+\beta_0)}}\right)$$

$$= -y_i \cancel{\log}\left((\beta_1 x+\beta_0) - \log\left(1+e^{(\beta_1 x+\beta_0)}\right)\right)$$

$$-(1-y_i)\left(\log\left(e^{-\beta_1 x+\beta_0}\right) - \log\left(1+e^{-(\beta_1 x+\beta_0)}\right)\right)$$

$$= -y_i(\beta_1 x+\beta_0) + y_i \log\left(1+e^{(\beta_1 x+\beta_0)}\right)$$

$$+(1-y_i)(\beta_1 x+\beta_0) + (1-y_i)\log\left(\frac{1+e^{\beta_1 x+\beta_0}}{e^{\beta_1 x+\beta_0}}\right)$$

$$-\sum \left(-y_i(\beta_1 x_i + \beta_0) - \log(1 + e^z) \right)$$

$$\begin{bmatrix} \dfrac{\partial}{\partial \beta_1} \\ \dfrac{\partial}{\partial \beta_0} \end{bmatrix}$$

$$z = \beta_1 x_i + \beta_0$$

$$\sum_{i=1}^{N} \begin{bmatrix} y_i x_i - \dfrac{1}{1+e^z} \cdot e^z \cdot x_i \\ y_i - \dfrac{1}{1+e^z} \cdot e^z \cdot \end{bmatrix}$$

$$-\sum_{i=1}^{N} \begin{bmatrix} x_i \left(y_i - \dfrac{e^{\beta_1 x_i + \beta_0}}{1+e^{\beta_1 x_i + \beta_0}} \right) \\ y_i - \dfrac{e^{\beta_1 x_i + \beta_0}}{1+e^{\beta_1 x_i + \beta_0}} \end{bmatrix}$$

$$-\sum_{i=1}^{N} \begin{bmatrix} x_i \left(y_i - \dfrac{1}{1+e^z} \cdot e^z \right) \\ y_i - \dfrac{1}{1+e^z} \cdot e^z \end{bmatrix}$$

$$-\sum_{i=1}^{N} \begin{bmatrix} (y_i - \sigma(x_i, \beta_1, \beta_0))\, x_i \\ y_i - \sigma(x_i, \beta_1, \beta_0) \end{bmatrix}$$

$$= y_i(\beta_1 x + \beta_0) + y_i \log\left(1 + e^{\beta_1 x + \beta_0}\right)$$

$$+ (1-y_i)(\beta_1 x + \beta_0) - (1-y_i)(\beta_1 x + \beta_0) +$$

$$(1-y_i)\log\left(1 + e^{\beta_1 x + \beta_0}\right)$$

$$= -\left(y_i(\beta_1 x + \beta_0) - \log\left(1 + e^{\beta_1 x + \beta_0}\right)\right)$$

Logistic Regression and Cross entropy Loss

$$\underset{(\beta_1, \beta_0)}{\dot{M}in} \; -\sum_{i=1}^{l} \left(y_i (\beta_1 x_i + \beta_0) - \; log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right)$$

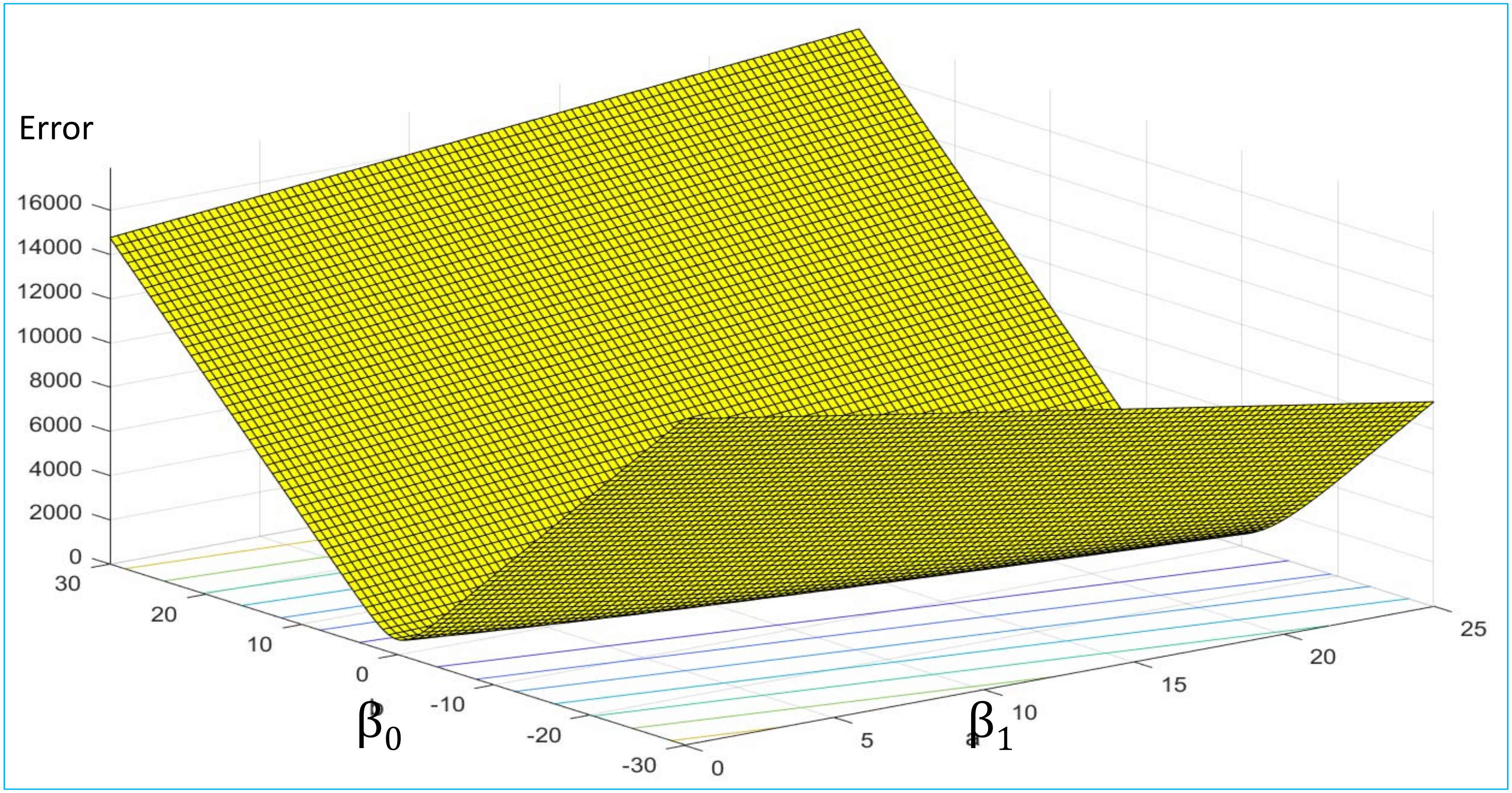
- Only minimization of $-\sum_{i=1}^{l} \left(y_i (\beta_1 x_i + \beta_0) - \; log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right)$ is not enough.
- Our estimated logistic function should not increase sharply as well.

Logistic Regression Optimization Problem

$$\underset{(\beta_1, \beta_0)}{\dot{M}in} J(\beta_1, \beta_0) = - \sum_{i=1}^{l} \left(y_{i}(\beta_1 x_i + \beta_0) - \log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1{}^2$$

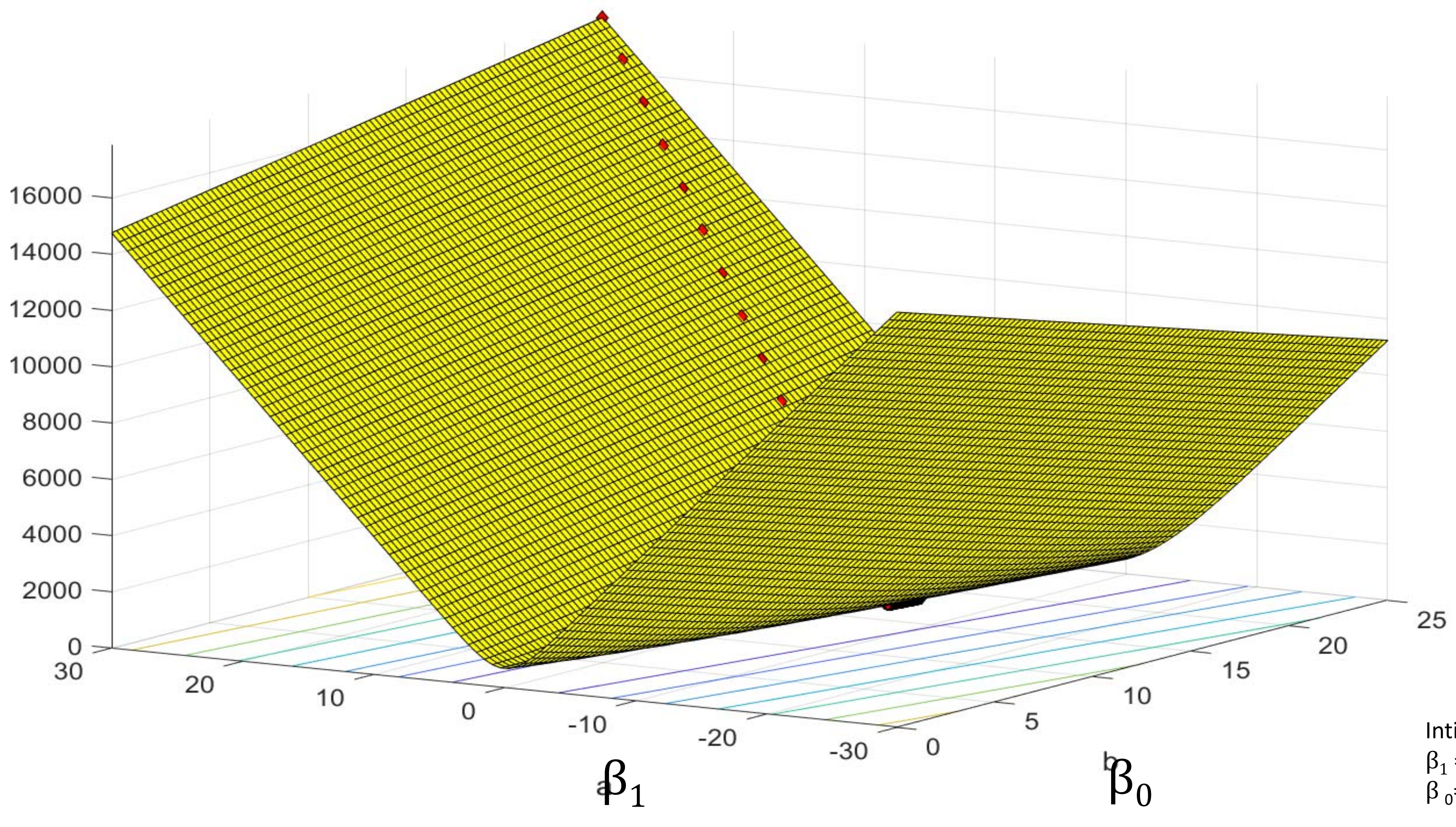
Empirical Loss

Regularization with parameter λ



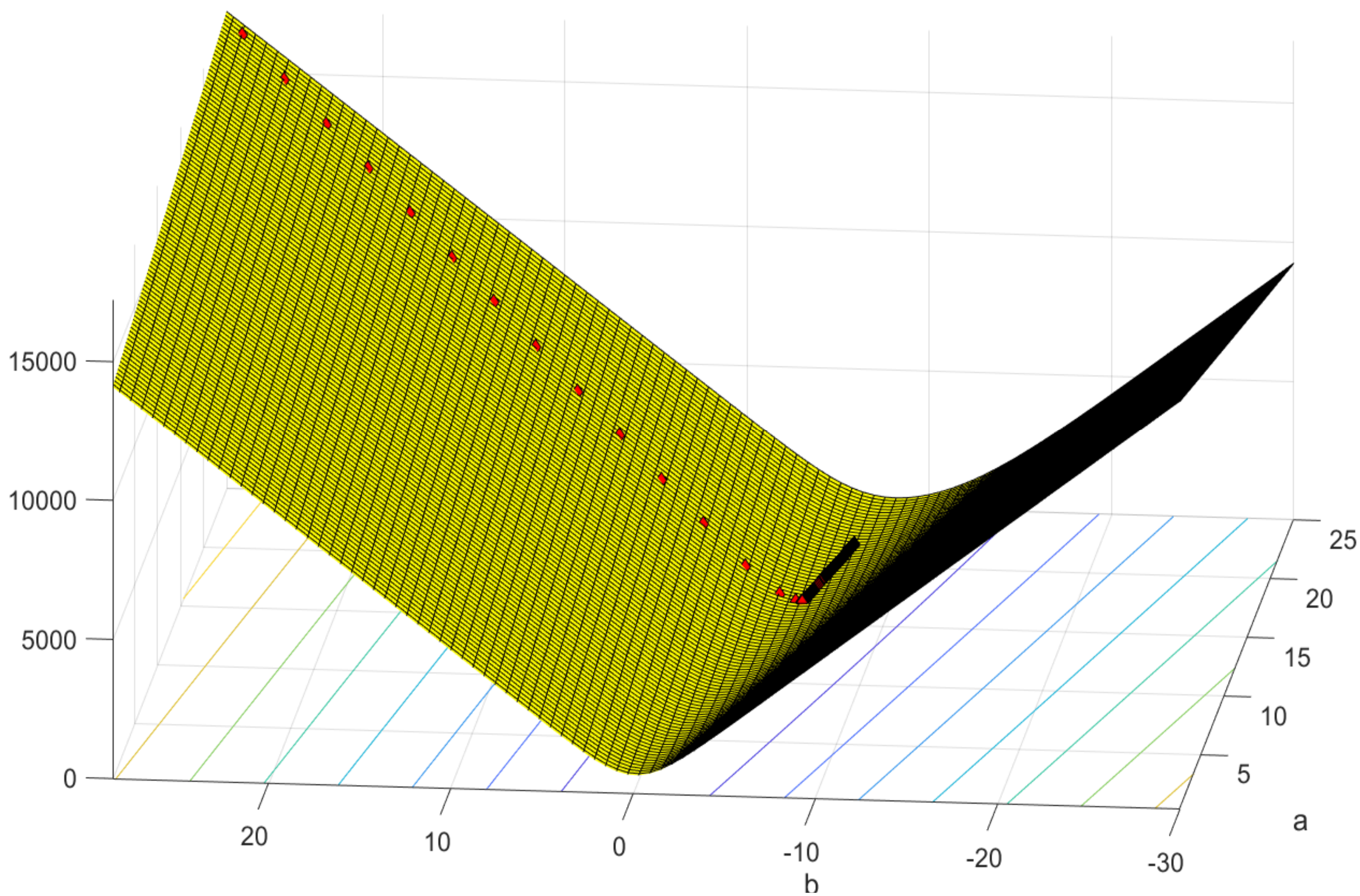
Error
16000
14000
12000
10000
8000
6000
4000
2000
0
30
20
10
0
-10
-20
-30
β_0
0
5
10
15
20
25
β_1

Gradient Descent Method



Intial values
β_1 = 25
β_0=30

Gradient Descent Method



Computing Gradients

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l} \left(y_i(\beta_1 x_i + \beta_0) - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1^{2}$$

Computing Gradients

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l}\left(y_{i(}\beta_1 x_i + \beta_{0)} - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1{}^2$$

$$\nabla_{\beta 1} J(\beta_1, \beta_0) = \lambda\, \beta_1 - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right))\, x_i$$

Computing Gradients

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l}\left(y_{i}(\beta_1 x_i + \beta_0) - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1{}^2$$

$$\nabla_{\beta 1} J(\beta_1, \beta_0) = \lambda\, \beta_1 - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right))\, x_i$$

$$= \lambda\, \beta_1 - \sum_{i=1}^{n}(yi - \sigma(x, \beta_1, \beta_0))xi$$

Computing Gradients

$\beta_1^T x_i + \beta_0$ $\beta_1^T x_i + \beta_0$

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l}\left(y_i(\beta_1 x_i + \beta_0) - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1^2$$

$$\nabla_{\beta 1} J(\beta_1, \beta_0) = \lambda\,\beta_1 - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right))\,\cancel{x_i}$$

$\beta_1^T x_i + \beta_0$

$$\frac{1}{1 + e^{-(\beta_1^T x + \beta_0)}}$$

$$= \lambda\,\beta_1 - \sum_{i=1}^{n}(yi - \sigma(x, \beta_1, \beta_0))xi$$

$$\nabla_{\beta 0} J(\beta_1, \beta_0) = -\sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right))$$

$\beta_1^T x_i + \beta_0$

$$= -\sum_{i=1}^{n}(yi - \sigma(x, a^{(k)}, b^{(k)}))$$

Gradient Descent Algorithm for Logistic Regression

Algorithm:- Gradient descent method

Initialize β_1^0 and β_0^0

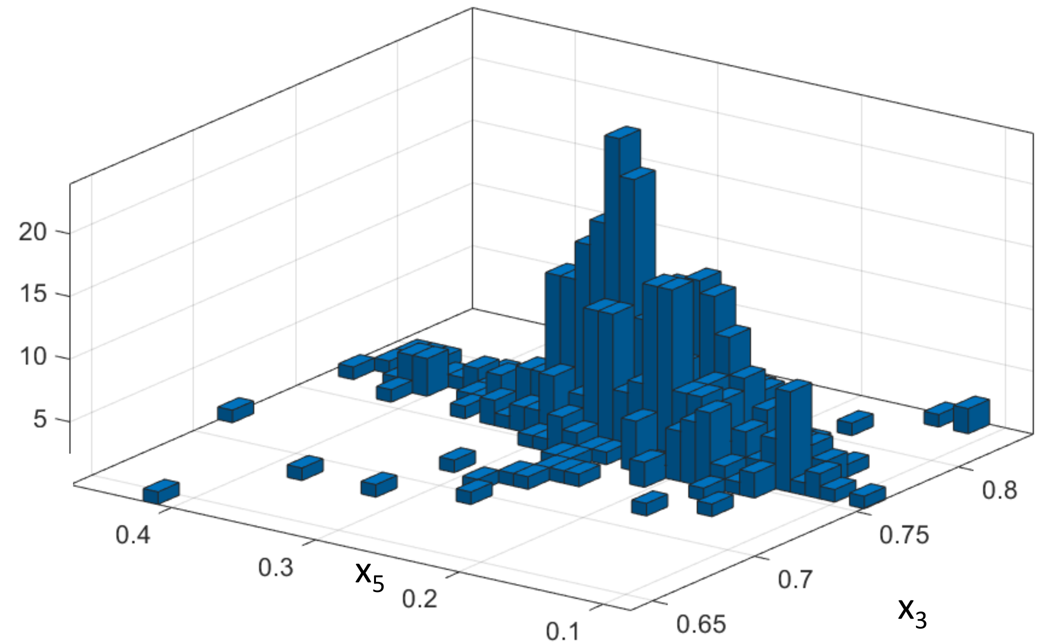
Repeat

$$\beta_1^{(k+1)} := \beta_1^{(k)} - \eta \nabla_{\beta 1} J(\beta_1^{(k)}, \beta_0^{(k)})$$

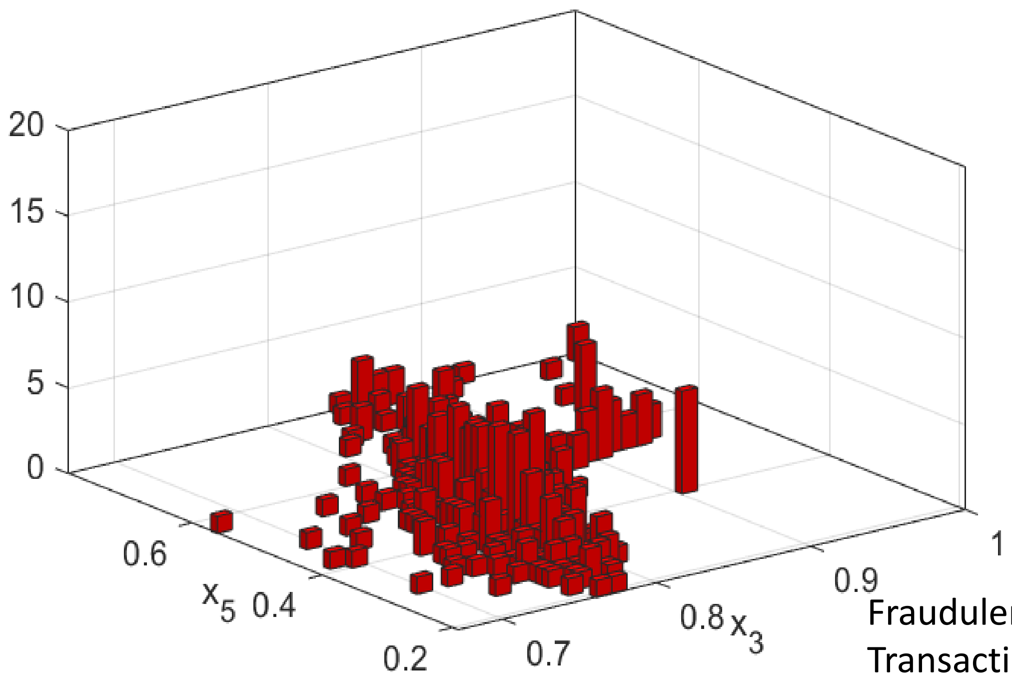
$$\beta_0^{(k+1)} := \beta_0^{(k)} - \eta \nabla_{\beta 0} J(\beta_1^{(k)}, \beta_0^{(k)})$$

Until $\left|\left| \begin{bmatrix} \nabla_{\beta 1} J(\beta_1^{(k+1)}, \beta_0^{(k+1)}) \\ \nabla_{\beta 0} J(\beta_1^{(k+1)}, \beta_0^{(k+1)}) \end{bmatrix} \right|\right| \leq \varepsilon$

Working with bivariate data

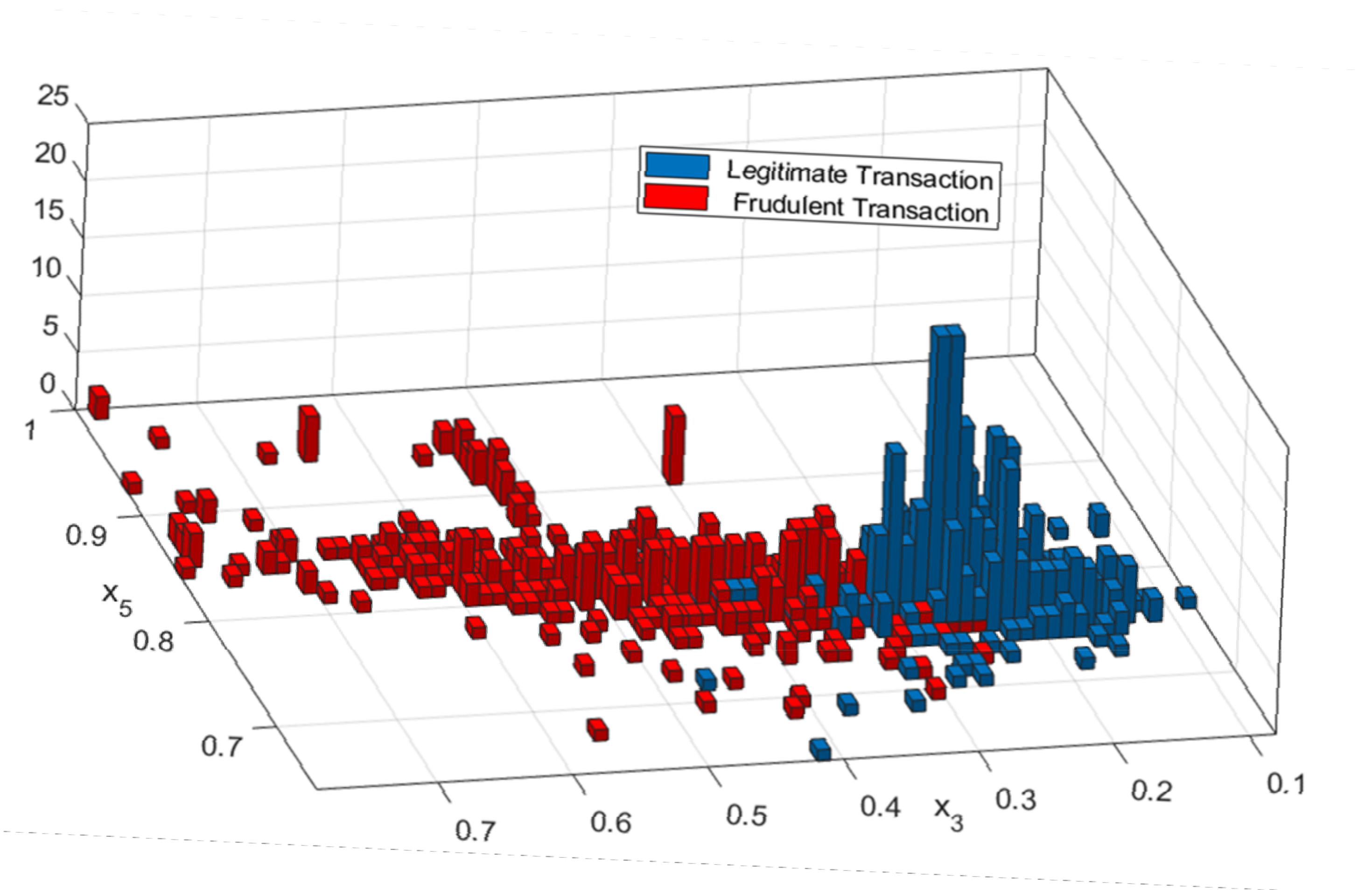


Legitimate Transaction



Fraudulent Transaction

Working with bivariate data

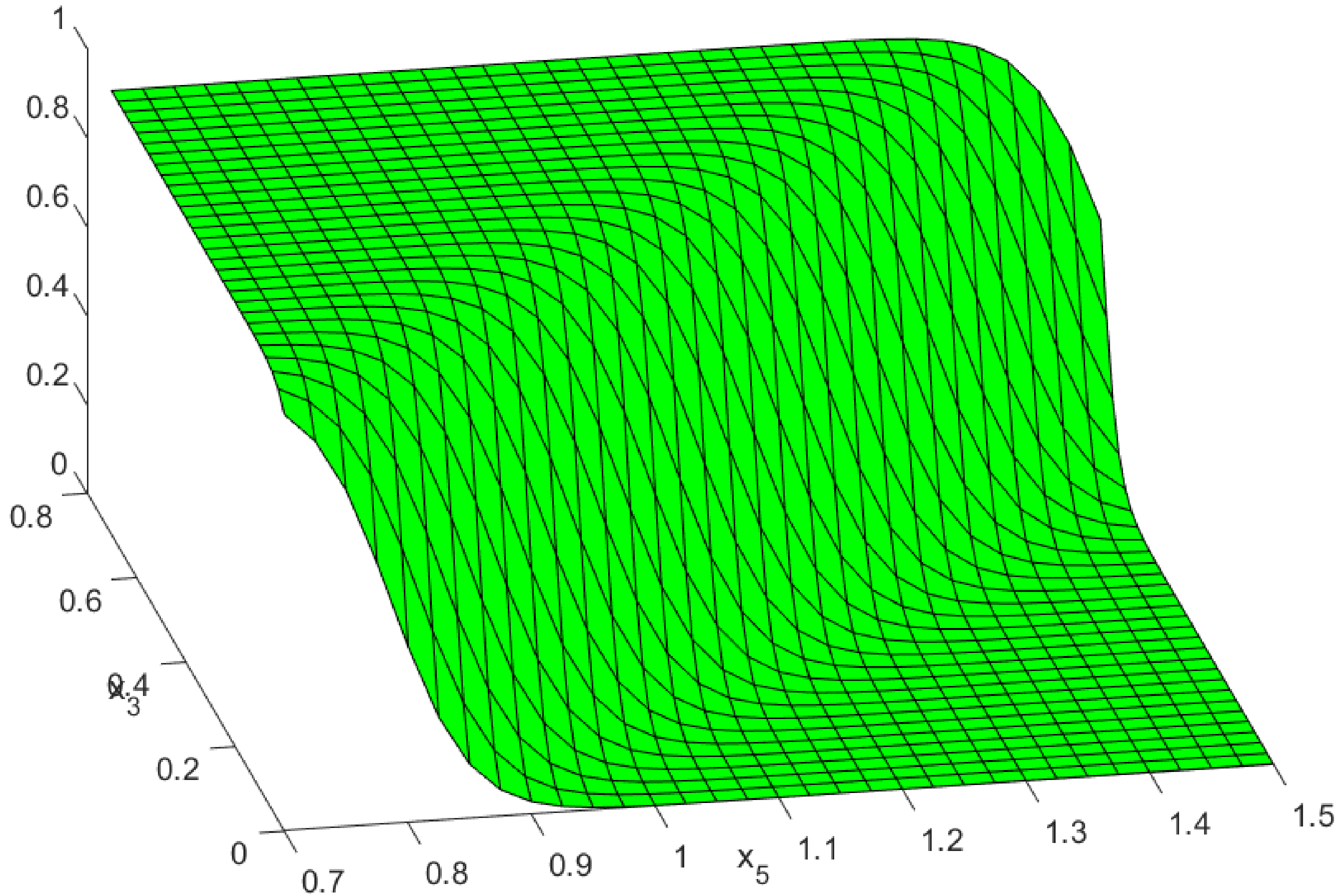


Logistic Regression for bivariate data

$$\frac{1}{1+e^{-(\beta_1^T \mathbf{x} + \beta_0)}}$$

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

https://www.desmos.com/calculator/coknirwubg

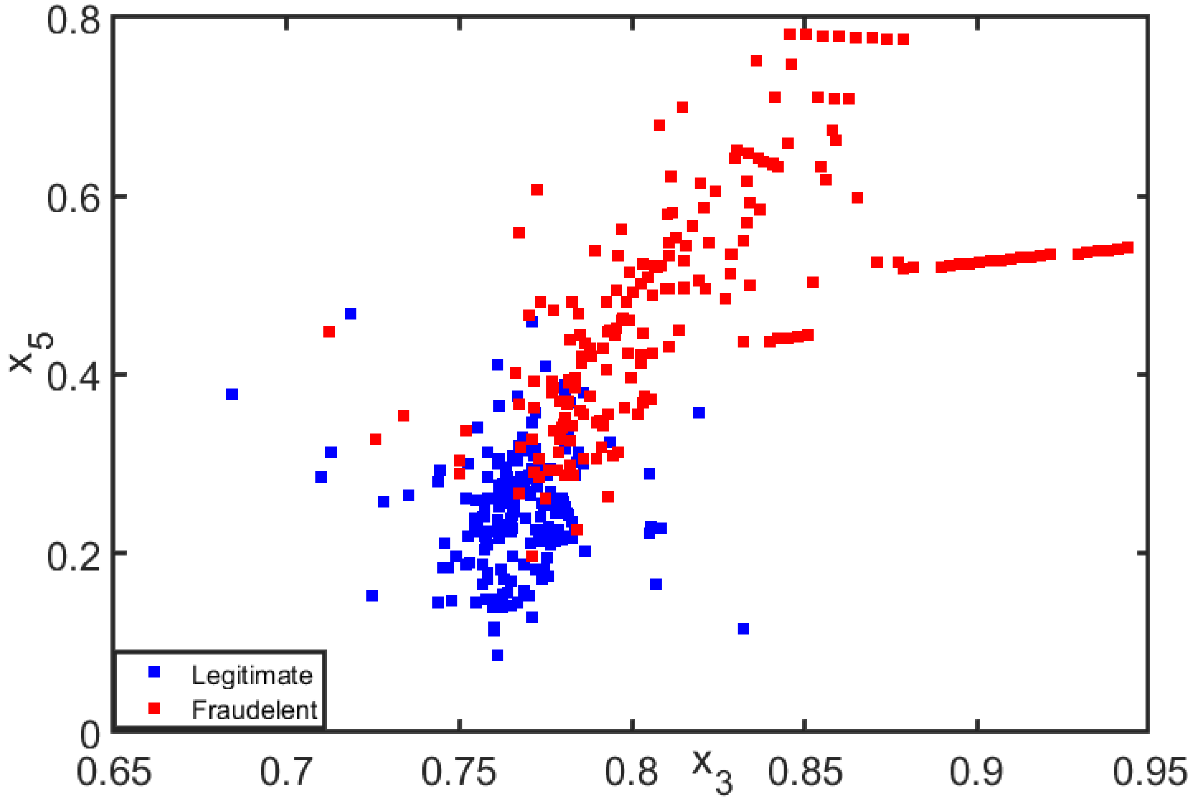


1
0.8
0.6
0.4
0.2
0
0.8
0.6
0.4
x_3
0.2
0
0.7
0.8
0.9
1
x_5
1.1
1.2
1.3
1.4
1.5

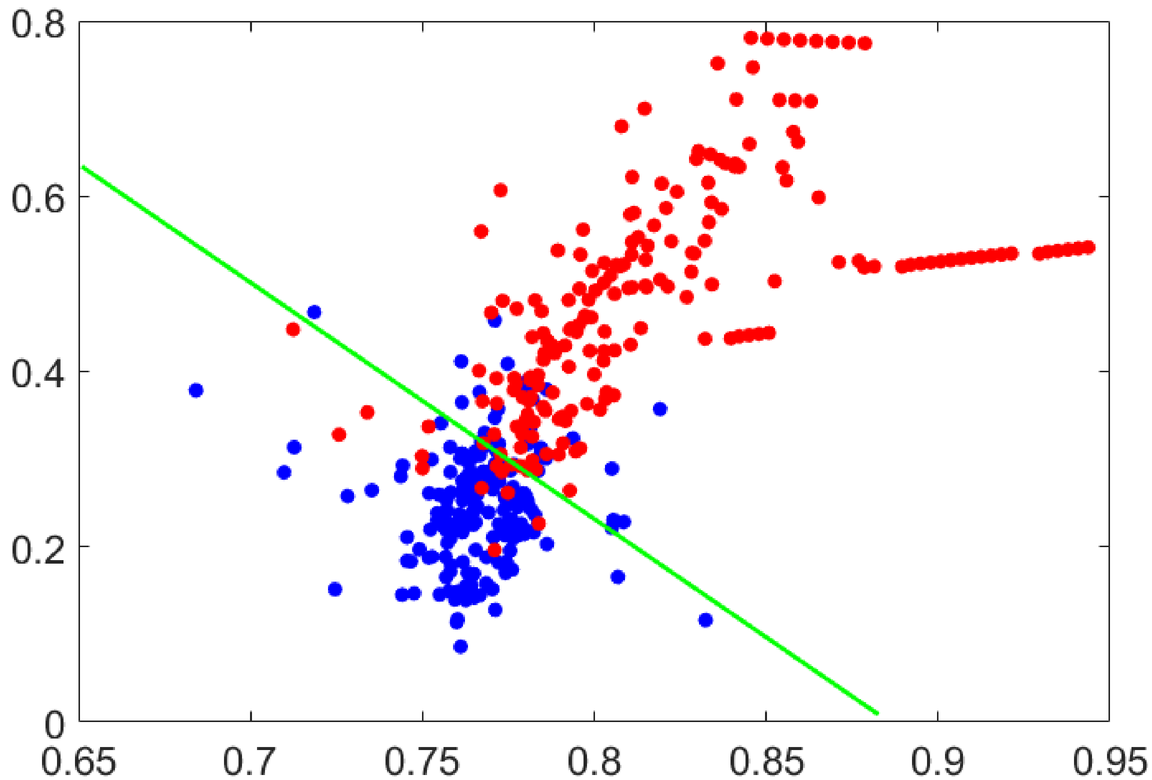
One more Observation

$$\frac{1}{1+e^{-(\beta_1 x+\beta_0)}} = 0.5$$

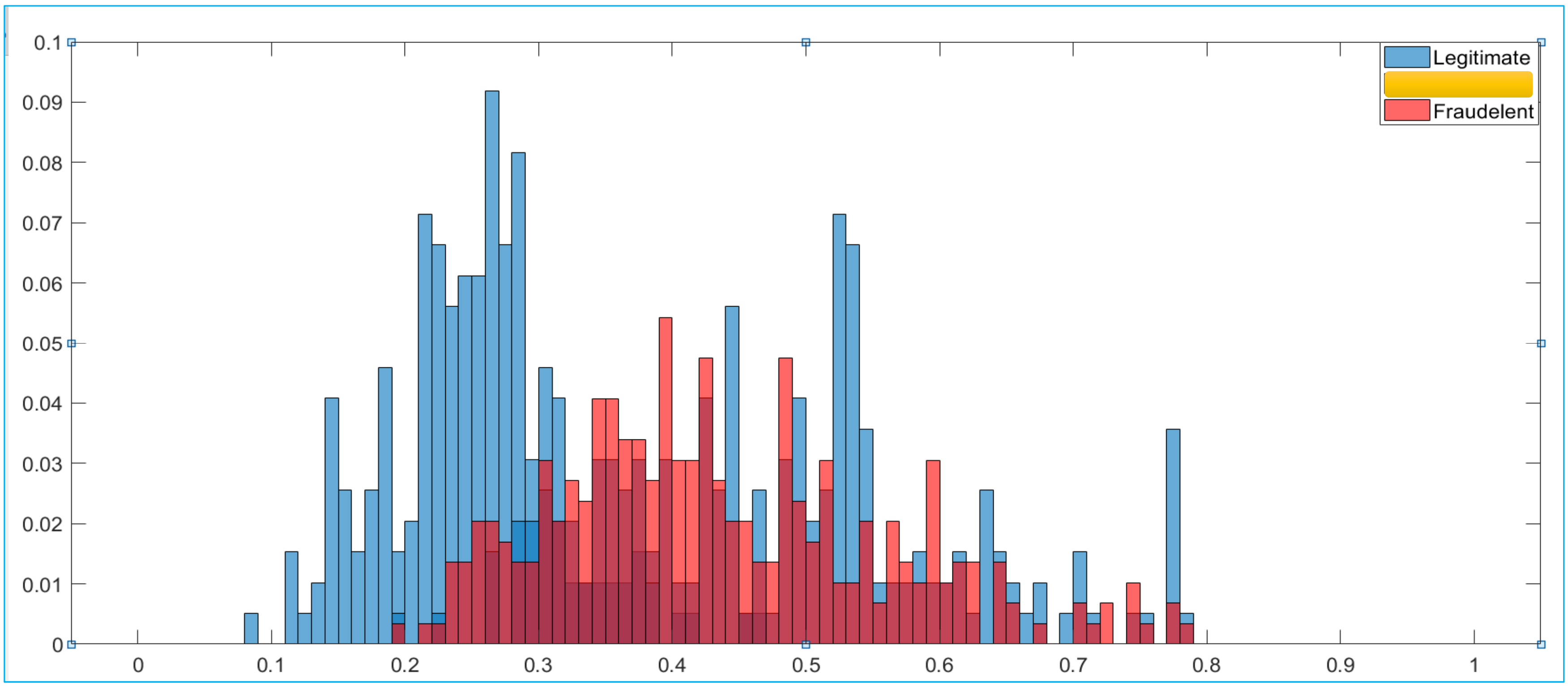
$$\beta_1 x + \beta_0 = 0$$

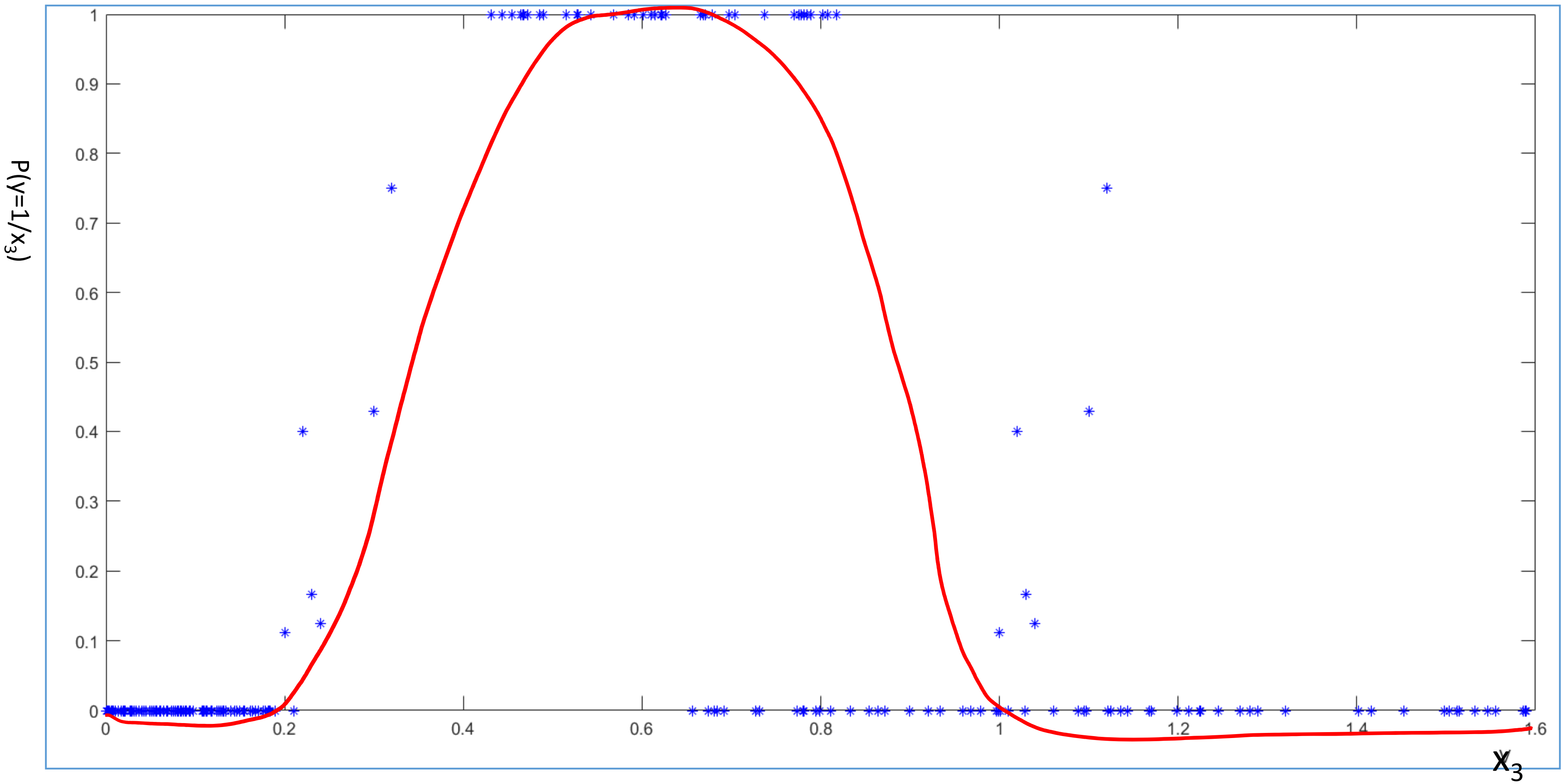


$$1 + \frac{1}{e^{5}}$$


0.8
0.6
0.4
0.2
0
0.65
0.7
0.75
0.8
0.85
0.9
0.95

Need for non-linear logistic regression





P(y=1/x3)
x3
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

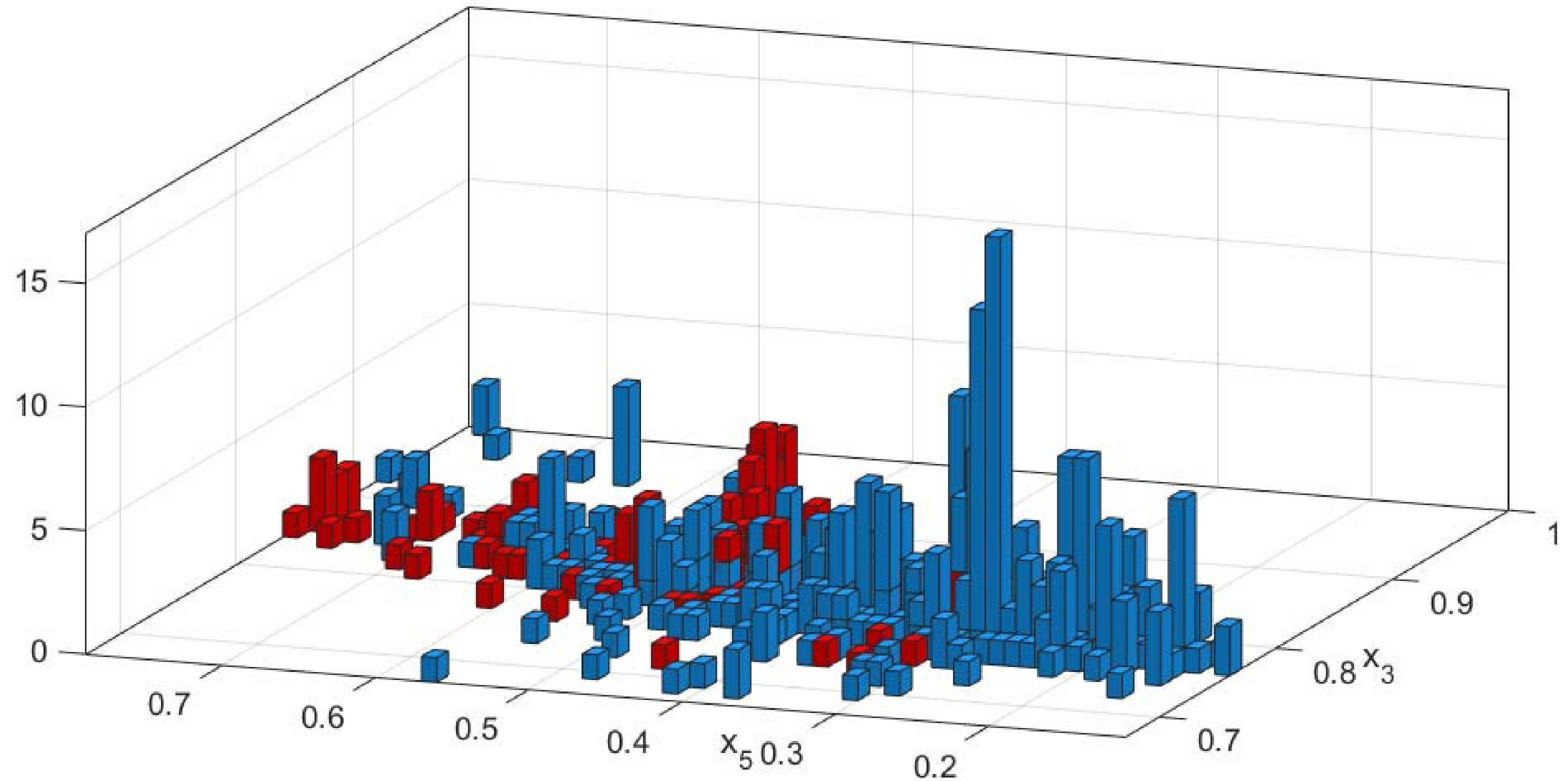
Non-linear Logistic Regression

$$\frac{1}{1+e^{-(\beta_1 x + \beta_0)}}$$

$$f(x) = \frac{1}{1+e^{-(\beta_2 x^2+\beta_1 x+ \beta_0)}} = \frac{e^{(\beta_2 x^2+\beta_1 x+ \beta_0)}}{1+e^{(\beta_2 x^2+\beta_1 x+ \beta_0)}}$$

https://www.desmos.com/calculator/coknirwubg

Non-linear logistic Regression for bivariate data



$$\frac{e^{\beta_1 x_1 + \beta_2 x_2 + \beta_0}}{1+e^{\beta_1 x_1 + \beta_2 x_2 + \beta_0}}$$

$$\frac{e^{\beta_1^T \phi(x) + \beta_0}}{e^{\beta_1^T \phi(x) + \beta_0}}$$

$$\frac{e^{\beta_1^T x + \beta_0}}{1+e^{\beta_1^T x + \beta_0}}$$

$$\phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 0 \end{bmatrix}$$

$$f(x) = \frac{1}{1+e^{-(\beta_1 x_1^{\ 2} + \beta_2 x_2^{\ 2} + \beta_3 x_1 x_2 + \beta_4 x_1 + \beta_5 x_2 + \beta_6)}}$$

$$= \frac{e^{(\beta_1 x_1^{\ 2} + \beta_2 x_2^{\ 2} + \beta_3 x_1 x_2 + \beta_4 x_1 + \beta_5 x_2 + \beta_6)}}{1+e^{(\beta_1 x_1^{\ 2} + \beta_2 x_2^{\ 2} + \beta_3 x_1 x_2 + \beta_4 x_1 + \beta_5 x_2 + \beta_6)}}$$

https://www.desmos.com/calculator/coknirwubg

$x_1 , x_2 \ldots x_n \quad y$

$X_i \in R^n$

$$\sigma(\beta_1, \beta_0, x) = \frac{1}{1+e^{-(\beta_1^T x + \beta_0)}}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\sigma(\beta_1, \beta_0, x) = \frac{1}{1+e^{-(\beta_1^T \phi(x) + \beta_0)}}$$

$$\beta_1 = \begin{bmatrix} \beta_{11} \\ - \\ - \\ - \\ \beta_{1n} \end{bmatrix} \qquad \beta_0 \in R$$

Consider the NAND dataset. Which one of the following may be the solution of logistic regression model ?

(a) $\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} -80.26 \\ -110.85 \\ 21 \end{bmatrix}$

(b) $\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} 80.26 \\ 110.85 \\ -152.61 \end{bmatrix}$

(c) $\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} -80.26 \\ 0 \\ 152.61 \end{bmatrix}$

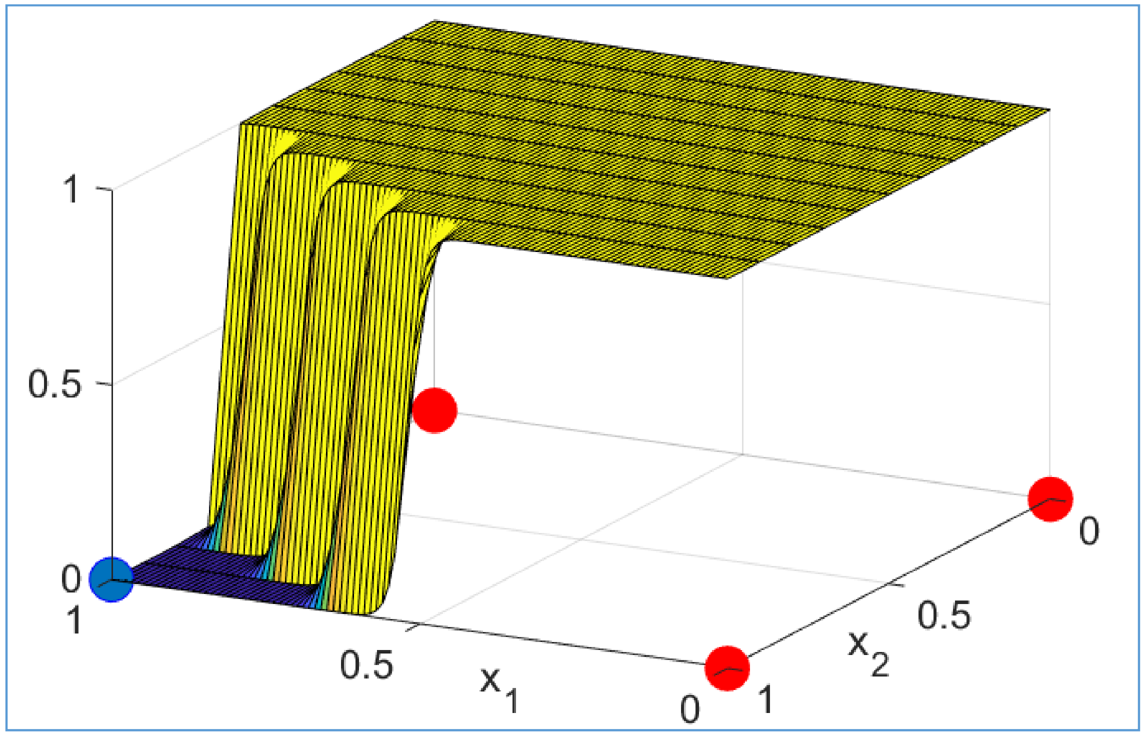
(d) $\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} -80.26 \\ -110.85 \\ 152.61 \end{bmatrix}$

$\beta_1 x_1 + \beta_2 x_2 + \beta_0 > 0$

$\hookrightarrow 1$

x1	x2	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND Dataset



1
0.5
0
1
0.5
x_1
0
1
x_2
0.5
0

$$T = \{ (x_1, y_1), (x_2, y_2) \ldots , (x_L, y_L) \}$$

$$\max \underline{P}(y_1, y_2, \; y_L / x_1, x_2, \; x_L)$$

$$= \max_{i=1} \prod_{i=1}^{L} \underline{P}(y_i / x_i)$$

$$y_i / x_i \sim B(P_{x_i})$$

$$= \max \sum_{i=1}^{L} \log \underline{P}(y_i / x_i)$$

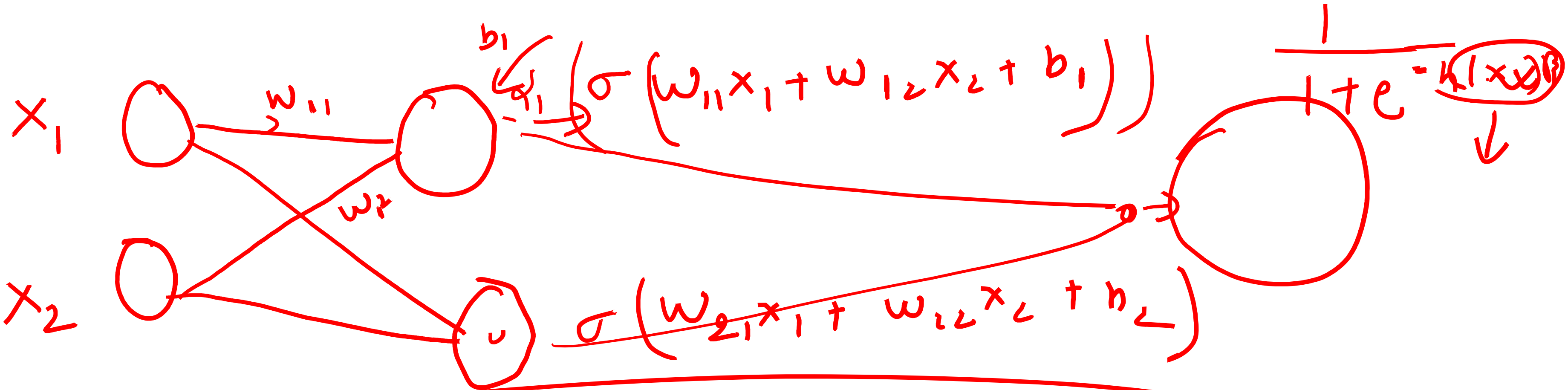
$$y_i = 1 / x_1 = \hat{P}_{x_1}$$

$$y = 0 / x_1 = 1 - \hat{P}_{x_1}$$

$$= \max \sum_{i=1}^{L} \log \left(\hat{P}_{x_i}^{y_i} (1 - \hat{P}_{x_i})^{1-y_i} \right) = \hat{P}_{x_1}^{y_i} (1 - \hat{P}_{x_1})^{1-y_i}$$

$$= \max \sum_{i=1}^{L} \left(y_i \, \log \hat{P}_{x_i} + (1 - y_i) \log (1 - \hat{P}_{x_i}) \right)$$

$$\min_{\underline{\underline{f}}} \sum_{i=1}^{l} -y_i \log f(x_i) - (1-y_i) \log (1 - f(x_i))$$



$$\alpha_1 \ \sigma_1(w_{11}x_1 + w_{12}x \ b_1) + \alpha_2(\sigma(w_{21}x_1 + w_{22}x_2 + b_2))$$

Linear Logistic Regression with k variables

$$f(x) = \frac{1}{1+e^{-(\boldsymbol{\beta}^T \boldsymbol{x}+\beta_0)}} = \frac{e^{(\boldsymbol{\beta}^T x+\boldsymbol{\beta}_0)}}{1+e^{(\boldsymbol{\beta}^T x+\boldsymbol{\beta}_0)}}$$

where $\beta = \begin{bmatrix} \beta_k \\ . \\ . \\ \beta_1 \end{bmatrix}$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Error = 1- Accuracy

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	70	30
Legitimate Transaction	100	1800

Accuracy = ??

Error = ??

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

What proportion of positive identifications was actually correct?

$$\text{Precision} = \frac{TP}{TP+FP}$$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	70	30
Legitimate Transaction	100	1800

Precision = ??

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

What proportion of actual positives was identified correctly?

Recall = $\frac{TP}{TP+FN}$

Also called sensitivity some time.

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	70	30
Legitimate Transaction	100	1800

Recall = ??

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

Specificity $= \frac{TN}{TN+FP}$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	70	30
Legitimate Transaction	100	1800

Specificity = ??

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	0	10
Legitimate Transaction	190	1800

Accuracy = ??

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

F1 score = $\frac{2 \times Precision \times Recall}{Precision + Recall}$

Gmean $= \sqrt{\text{Specificity} \times Recall}$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	True Positive	False Positive
Legitimate Transaction	False Negative	True Negative

F1 score = $\frac{2 \times Precision \times Recall}{Precision + Recall}$ Gmean $= \sqrt{\text{Specificity} \times Recall}$

Some notes on Evaluation Metric

	Predicted Fraudulent	Predicted Legitimate
Fraudulent Transaction	0	10
Legitimate Transaction	190	1800

F1 Score ??
G-mean ??

Gradient Descent Algorithm for large scale datasets

Algorithm:- Gradient descent method

Initialize β_1^0 and β_0^0

Repeat

$$\beta_1^{(k+1)} := \beta_1^{(k)} - \eta \nabla_{\beta 1} J(\beta_1^{(k)}, \beta_0^{(k)})$$

$$\beta_0^{(k+1)} := \beta_0^{(k)} - \eta \nabla_{\beta 0} J(\beta_1^{(k)}, \beta_0^{(k)})$$

$$\text{Until } \left|\left| \begin{bmatrix} \nabla_{\beta 1} J(\beta_1^{(k+1)}, \beta_0^{(k+1)}) \\ \nabla_{\beta 0} J(\beta_1^{(k+1)}, \beta_0^{(k+1)}) \end{bmatrix} \right|\right| \leq \varepsilon$$

Gradients for large-scale datasets

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l}\left(y_{i(}\beta_1 x_i + \beta_{0)} - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1{}^2$$

$$\nabla_{\beta 1} J(\beta_1, \beta_0) = \lambda \beta_1 - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right)) x_i$$

$$= \lambda \beta_1 - \sum_{i=1}^{n}(yi - \sigma(x, \beta_1, \beta_0))xi$$

$$\nabla_{\beta 0} J(\beta_1, \beta_0) = - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right))$$

$$= - \sum_{i=1}^{n}(yi - \sigma(x, a^{(k)}, b^{(k)}))$$

Stochastic gradients for large-scale datasets

$$J(\beta_1, \beta_0) = -\sum_{i=1}^{l}\left(y_i(\beta_1 x_i + \beta_0) - log\left(1 + e^{(\beta_1 x_i + \beta_0)}\right)\right) + \frac{\lambda}{2}\beta_1^{2}$$

$$\nabla_{\beta 1}\ J(\beta_1, \beta_0) = \lambda\,\beta_1 - \sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right)\ )\ x_i$$

$$= \lambda\,\beta_1 - \sum_{i=1}^{n}(yi - \sigma(x, \beta_1, \beta_0)\ )xi$$

$$\nabla_{\beta 0}\ J(\beta_1, \beta_0) = -\sum_{i=1}^{n}(yi - \left(\frac{1}{1+e^{-(\beta_1 x_i + \beta_0))}}\right)\ )$$

$$= -\sum_{i=1}^{n}(yi - \sigma(x, \mathrm{a}^{(k)}, b^{(k)}))$$