Logistic Regression

Till now ml → supervised → Regression → SLR, MLR
                          → Classification → Logistic Reg.
           → unsupervised

eg1 → ~~#~~ no. of hours studied → Predict Pass/fail

Hours Studied	Pass/Fail
1	F
2	P
3	P
1	P
2	F
4	P
5	P
1	F
2	F
1	F
4	P
3	F

What we want ↓ train

new data → [Model] P/F or (1/0)

Accuracy ↑

This comes under classification problem

eg2 → Predict weather person frauds / not frauds (bank

eg3 → Predict weather mail is spam / ham

classification

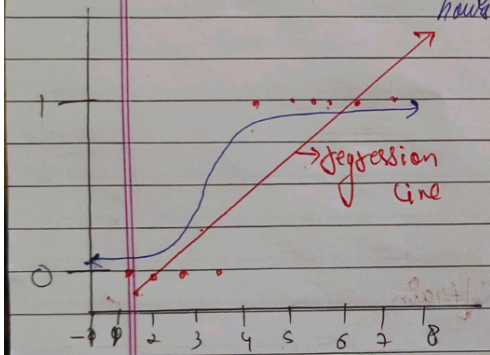
Binary → output → 2 categories
↓
Status → Fail
Pass

Multi Class
more than 2 categories
↓
Risk → high
medium
low

There are different algorithm to solve classification problem.

1 → Logistic Regression

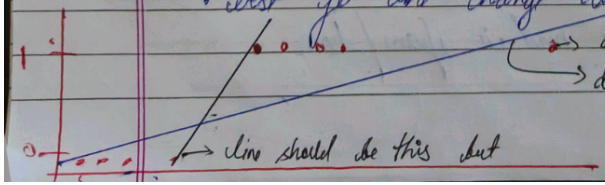
x (no of hours studied)	y (status P/F)
2	F
0	F
3	P
4	P
1	F
6	P

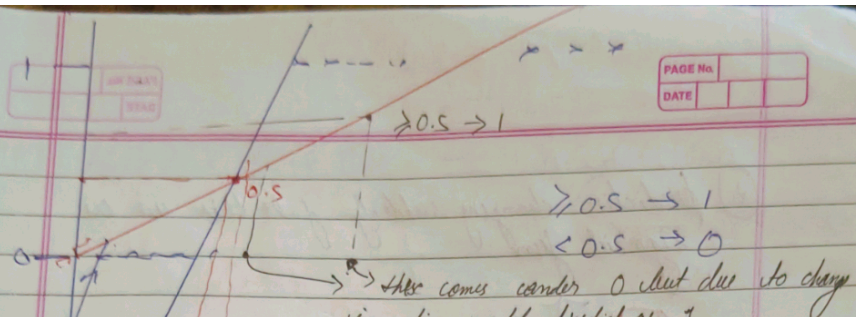


→ Can we solve it using regression line?

No → • because Nature of problem statemen is diff. in regression target value should continous here it is discrete

• best fit line change due to presence of outlier





3 → Related to first point

↳ range of regression line → (-∞, +∞)

" " classification " → [0, 1]

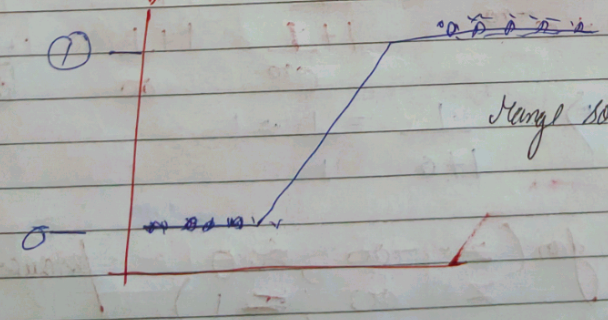
- prediction changes rapidly / suddenly

↳ in above eg 2.3 → Pass / 2.2 → fail

↳ this is not meaningful in real life, it should change gradually.

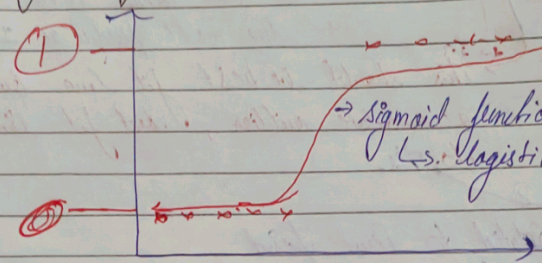
How to Solve the problem?

① Squeezing of line to [0, 1]



PAGE No.
DATE

① predictin changing suddenly for this we use sigmoid fun



formulae for this fun $= \frac{1}{1+e^{-z}}$ → range [0, 1]

Proof: ↳ ① $z = -\infty$

$\frac{1}{1+e^{-(-\infty)}} = \frac{1}{1+e^{\infty}} = \frac{1}{1+\infty} = \frac{1}{\infty} \approx 0$

divide any 1 with $\infty \approx 0$

↳ any thing to power(∞) is (∞)

② $z = \infty$

$\frac{1}{1+e^{-\infty}} = \frac{1}{1+\frac{1}{e^{\infty}}} = \frac{1}{1+\frac{1}{\infty}} = \frac{1}{1+0} = 1$

$\frac{1}{\infty} \approx 0$

$\Rightarrow \frac{1}{1+0} = \frac{1}{1} = 1$

for ($z = -\infty, \approx 0$ $z = \infty, \approx 1$) proved.

① $H_\theta(x) = \theta_0 + \theta_1 x$ ⟶ best fit line

Now in logistic regression

$$H_\theta(x) = \frac{1}{1+e^{-z}} \qquad z = \theta_0 + \theta_1 x$$

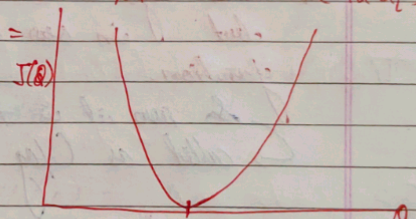
$$H_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$$ → Logistic regression model

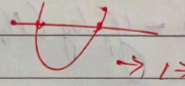
② To get best (θ_0, θ_1) we minimize cost function (CF)

CF of Logistic regression?

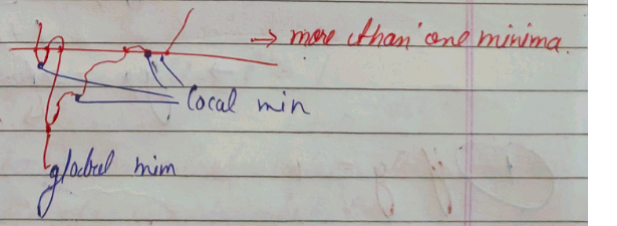
In SLRM → $CF = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$ → $(H_\theta(x_i))$

$J(\theta_0, \theta_1) =$



• Convex →  cuts at 2 point

→ 1 → minima, 1 → maxima

• non convex → 

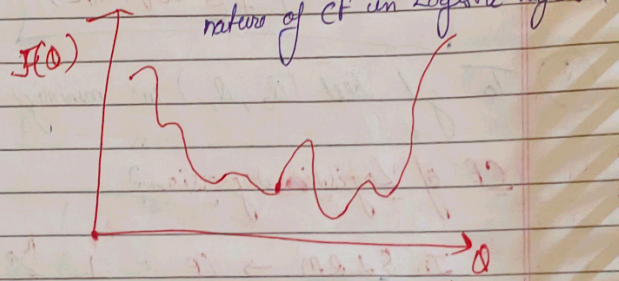
Log loss fun → $J(\theta_1,\theta_0) \Rightarrow -y_i \log(H_\theta(x_i)) - (1-y_i)(\log(1-H_\theta(x_i)))$

If $y=1 \Rightarrow J(\theta_1,\theta_0) = -y_i(\log(H_\theta(x_i)))$

In Logistic Regression

$$J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - H_\theta(x))^2$$

$$h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$$



In ML algo → convex cost fun is preferred but in our case we have non convex function.

↳ So now it is tweaked to get a new cost fun.

↳ called as (log loss function.)

log loss fun → $J(\theta_0,\theta_1) \rightarrow -y_i \log(H_\theta(x_i)) - (1-y_i)\log(1-H_\theta(x_i))$

→ $H_\theta(x_i) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$

→ y_i = actual value

↳ possible value → 0, 1

(1) if $y=1$

$$J(\theta_0,\theta_1) = -y_i \log(H_\theta(x_i)) - \cancel{(1-y_i)\log(1-H_\theta(x_i))}$$

$(1-1)=0$

$$J(\theta_0,\theta_1) = -y_i \log(H_\theta(x_i))$$

If $y=0$ $\quad J(\theta_0,\theta_1) = -\log(1-H_\theta(x_i))$

PAGE No.
DATE

If $y=0$

$$J(\theta_0,\theta_1) = -y_i \log(H_\theta(x_i)) - (1-y_i)(\log(1-H_\theta(x_i)))$$

($-y_i \log(H_\theta(x_i))$ → 0 as $y_i = 0$; $-(1-y_i)$ → 1 as $y_i = 0$)

$$J(\theta_0,\theta_1) = -(1)(\log(1-H_\theta(x_i)))$$

$$J(\theta_0,\theta_1) = -\log(1-H_\theta(x_i))$$

↳ To minimize CF we change $(\theta_0\ \theta_1)$ by using Convergence Algorithm

↳ Repeat until convergence

$$\{\ \theta_j := \theta_j - \alpha \frac{\partial J(\theta_0,\theta_1)}{\partial \theta}\ \}$$

↳ To get optimal $\theta_0\ \theta_1$

For one Independent variable

for multiple IV $\quad H_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\theta_2 x_2 \cdots \theta_n x_n)}}$

IMP Note → Logistic function range → (0 to 1)

Output of regression model → $P = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$

↓

between (0 to 1) Some Prob. value then we take cut off as 0.5 to put them in class(0 or 1)

PAGE No.
DATE

Logistic Regression with regularization

$$CF = J(\theta_0 \theta_1) = -y_i \log(H_\theta(x_i)) - (1-y_i)\log(1-H_\theta(x_i))$$

$$H_\theta(x_i) = \frac{1}{1+e^{-(\theta_0 + \theta_1 x_1)}}$$

1-> Ridge

$$J(\theta_0 \theta_1) = -y_i \log(H_\theta(x_i)) - (1-y_i)\log(1-H_\theta(x_i)) + L_2 \text{ regularization}$$

reduce overfitting

$$\lambda \sum_{i=1}^{n} \theta_i^2 \Leftarrow \lambda \theta_1^2$$

2-> Lasso

$$J(\theta_0 \theta_1) = -y_i \log(H_\theta(x_i)) - (1-y_i)\log(1-H_\theta(x_i)) + L_1 \text{ regularization}$$

feature selection.

$$\lambda |\theta_1| \Rightarrow \lambda \sum_{i=1}^{n} |\theta_i|$$

3-> Elastic Net

$$J(\theta_0 \theta_1) = CF + \underset{\text{Reg.}}{L_1} + \underset{\text{Reg.}}{L_2}$$

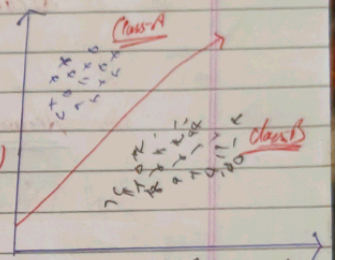
both reduce overfitting & feature selection

in logistic regression parameter we have $\left[C = \frac{1}{\lambda}\right]$

Classification Evaluation Metrics

→ Classification metrics
- Confusion matrix
- Accuracy / misclassification rate
- Precision
- Recall
- F-Beta score
- True Positive Rate (sensitivity)
- False Positive Rate
- True Negative rate (specificity)
- ROC-AUC
- ~~At~~ Precision-Recall

Widely used in industries



class A & B are seperable

① Confusion Metric

Based on some cutoff

by $H(x) = \frac{1}{1+e^{-(B_0+B_1x)}}$

x_1	x_2	y_{act}	y_{pred}	
–	–	0	1	→ False / wrong classification
–	–	1	1	→ True / correct "
–	–	1	0	→ wrong "
–	–	0	0	→ Correct "
–	–	1	1	→ correct "
–	–	0	0	→ correct "
–	–	0	1	→ wrong "

True Positive → $y_{act}=1$, $y_{pred}=1$
False Positive → $y_{act}=0$, $y_{pred}=1$
True Negative → $y_{act}=0$, $y_{pred}=0$
False Negative → $y_{act}=1$, $y_{pred}=0$

To make this in number of organizing we have confusion matrix

Predicted Value \ Actual	1	0
1	True positive	False positive
0	False negative	True negative

Total Prediction = TP + TN + FP + FN

	y act 0	y act 1
y pred 0	2	2
y pred 1	1	2 → TP + TN

(2) Accuracy ⇒ How many are correctly predicted from all data point

Accuracy = TP + TN / All prediction
= 2 + 2
= 4

Accuracy = TP + TN / TP + FP + TN + FN

= 4/7

opposite of Accuracy is misclassification ⇒ FP + FN / Total Prediction

⇒ FP + FN / TP + FP + TN + FN

⇒ 1 − Accuracy

(3) Precision

why precision?

Before going deep let's understand why precision.

1000 pairs of data → 900 → class 0
→ 100 → class 1

		y_test = 1	y_test = 0
confusion matrix →	y_pred = 1	0	0
	y_pred = 0	100	900

Here our model predicted all data points as zero, all class 1 are wrongly predicted. even after that due to the class imbalance our accuracy comes out to be 90%.

$$\text{Accuracy} = \frac{900}{1000} = 0.90$$ } but it doesn't make any sense as it doesn't predicted as 1 class.

for this, we have Precision or other classification metrics

(1) Precision ⇒ $\frac{TP}{TP+FP}$ { out of all actual values how many are correctly predicted

↳ for class 1, out of all predicted 1, How many are actually one.

(2) Recall = $\frac{TP}{TP+FN}$ ⇒ { out of all predicted value How many are correctly predicted

Note ⇒ Both Precision & Recall we have seen for Positive class (1). Similarly we can see for class (0). but we generally intrested in knowing of positive class.

Note → when FN is imp → Recall
when FP is imp → Precision

When to use Precision / Recall.

Use Cases → (1) Spam classification → Spam / ham
1 0

↳ Here what is imp | FP | FN | TP + TN ?

Ans ⇒ FP → because lets say i worked a job and selected at one company, they sent me offerletter but it went to spam. and i haven't checked spam folder. So Here we gave Priority to FP

Precision = $\frac{TP}{TP+FP}$ | Recall = $\frac{TP}{TP+FN}$

(2) → where FN is important

→ Person is diabetic or not Diabetic / Not Diabetic
1 0

- in this case when you visit doctor, he says you are not diabetic but you are, this can be harmful for you.
- and if you are non diabetic and he says you are, you might focus more on your health & diet.

Then we can say in this case, False Negative is important. Then Recall is used.

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

PAGE No.
DATE

(5) F - β Score

$$\frac{(1+(\beta)^2)\ Precision \times Recall}{Precision + Recall}$$

When False Positive & False Negative both are imp

Value of β for diff cases

(1) when both FP & FP are imp

$\beta = 1$, F_1 ~~Score~~ $= 2 \times \frac{P \times R}{P+R}$

(2) When FN is imp

$\beta = 0.5$, F_1 score $= 1 + 0.25 \times \frac{P \times R}{P+R}$

(3) when ~~FN is imp~~ FP is imp

$\beta = 2$, F_1 Score $= 1+$

(6) True Positive Rate

out of all 1, it is actually predicting 1

$TPR = \frac{TP}{TP+FN}$ } Also called Recall

(7) False Positive Rate

when it is 0, How often it is predicting it as 1

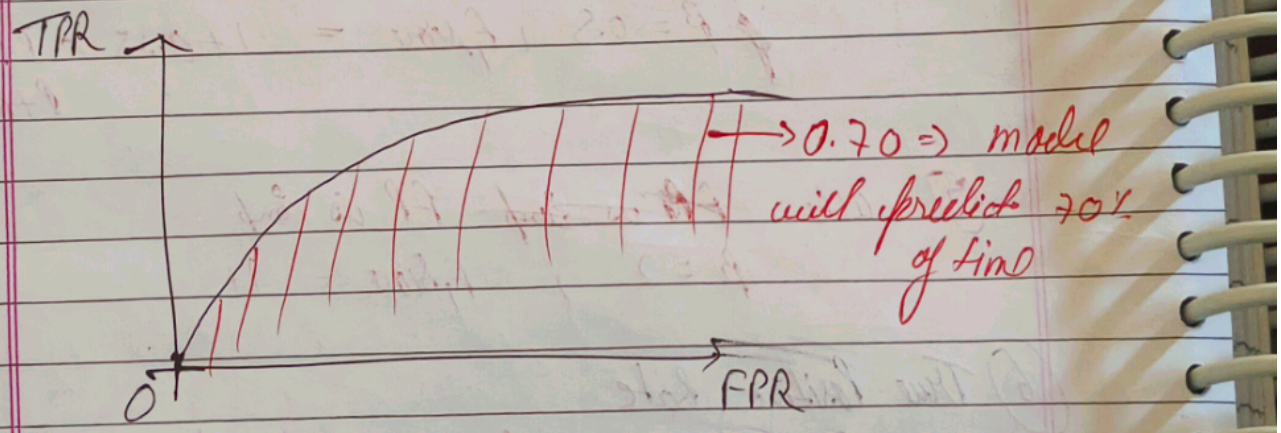
$FPR = \frac{FP}{TN + FP}$

8 True Negative Rate → out of all actual 0, it is actually predicted 0

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

$$TNR = 1 - FPR$$

9 AUR - ROC → Recives operating Characteristic - Area under curve



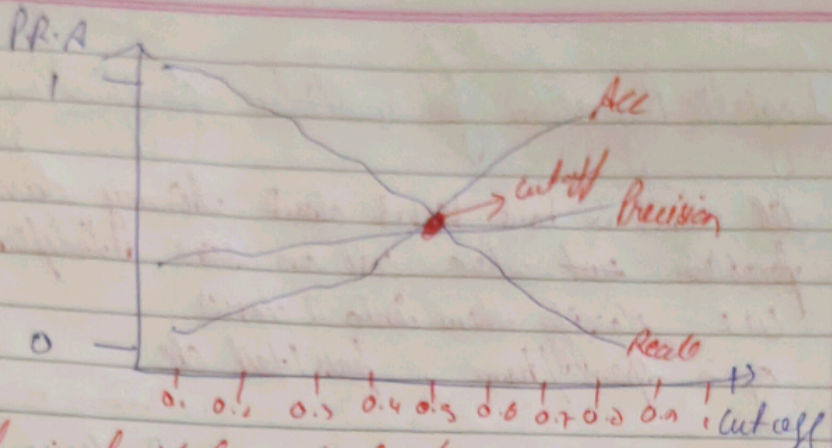
Higher the AUC → Better our model is

10-> Precision - Recall - Accuracy trade off

↳ when all three are imp (P, R, Acc)

How to decide ~~cut off~~ cutoff to classify the classes?

↳ P-R-C trade off is used for that



cutoff is decided as that point onto where all (P,R,A) three inter-sect each other.

U	y pred	Cutoff-0.1	. -	0.9

for all cut of we will calculate (P, R, A), then we plot then all.
Wherever they intersect this will be our point.

*** Logistic Regression for multiclass classification

till now we have learnt about binary classification problem where base on some independent variables we classify them into 2 classes.
eg → Spam / ham , pass / fail etc.

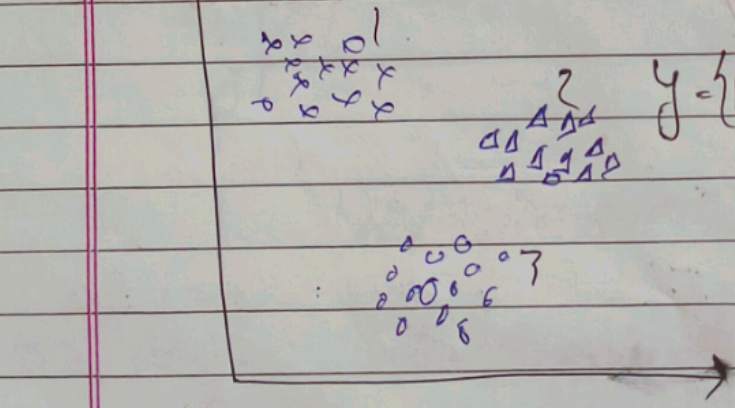
but when we have more than 2 classes it is called as multiclass classification.

eg → student grade → A, B, C, Fail

Age category → young, old, children

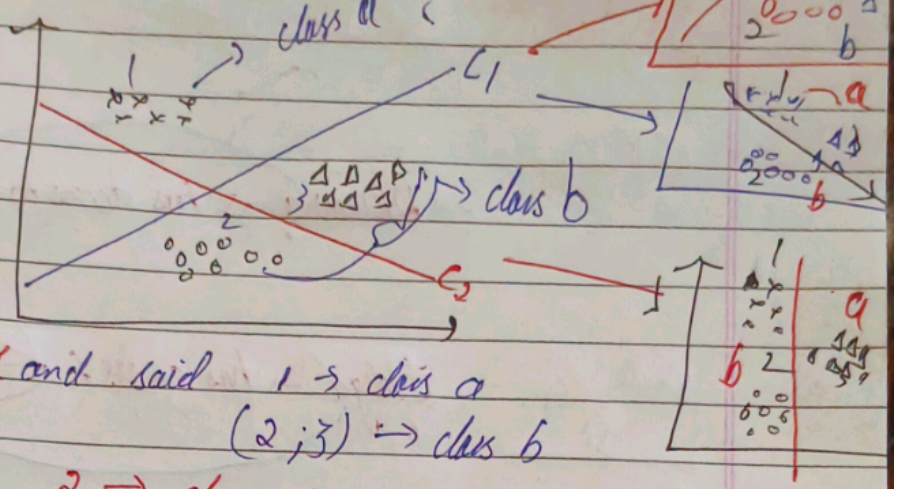
for them we have different algo.
Sol^n ⇒ OVR (One Vs Rest)
Multinomial (optional)

Current Scenario



(1) One Vs Rest → we will try to modify the logistic in such a way that we were able to solve multi class problem.

How to Solve it?



- Draw a line1 and said 1 → class a
  (2,3) → class b

  line 2 → 2 → class a
  (1,3) → class b

  line 3 → (1,2) → b
  (3) → a

Here No of logistic Regression model = no of classes but present

x_1	x_2	y	m_1 (class 1)	m_2 (class 2)	m_3 (class 3)
-	-	1	1	0	0
-	-	3	0	0	1
-	-	2	0	1	0
-	-	1	1	0	0
-	-	2	0	1	0
-	-	3	0	0	1

one vs Rest

m_1 = class1 Vs Rest

m_2 = class 2 vs Rest

m_3 = class 3 vs Rest

⇒ Now we have made the model but How will the prediction made?

Example → Virat kohli → has access → Batting Coach
→ Bowling Coach
→ Fielding Coach

Bhumra → has access to → Batting coach
→ Bowling Coach
→ Fielding Coach

Jadeja → has access to → Batting
→ Bowling
→ Fielding

Here all players goes to that coach who has highest knowledge about their skills.

Similarly in one vs Rest algo we have n no of models where n is the no. of classes

x_1	x_2	m_1	m_2	m_3
—	—	0.31	0.29	0.82
—	—	0.73	0.30	0.41
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

m_1 is specialist for class 1 → m_2 for class 2 → m_3 for class 3

Now lets say for $x_1 = 3$, $x_2 = 7$ we have 3 model with diff. prob.

x_1	x_2	m_1	m_2	m_3
3	7	0.31	0.27	0.87

m_1 is 31% confident x_1, x_2 belongs to class 1

m_2 is 27% confident

m_3 is 87% confident

now for x_1, y_1 we have m_3 is highest confident that it belongs to class 3. so we will choose it.

Note → Final prediction is that class/model which is highest confident about any data point.

Draw back → we were creating n models for 1 problem making it computationally expensive as binary classification is trained for each class

Step 1 → no of model = no of class

Step 2 → make model for each and get the probability for each model

Step 3 → Attach to that class which has highest probability.

→ This problem is solved ~~using~~ using a multinomial method / Softmax regression

- Multinomial method / Softmax regression
  → but we don't decompose the problem into binary classification instead we modify cost function.
  → Single model to reduce cost

Sigmoid (σ) $= \frac{1}{1+e^{-z}}$

Softmax ($\sigma(z_i)$) $= \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$

j - no of class

$z = (1, 2, 3)$

$$\sigma(z_1) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}, \quad \sigma(z_2) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\sigma(z_3) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\text{Cost fun} = \frac{-1}{n} \sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

→ after modification for the softmax

$$\text{Cost fun} = \frac{-1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} y_k^{(i)} \log \hat{y}_k^{(i)}$$

n = no of Data point
K = no of class.

features (x_1, x_2	y	$y_{k=1}$	$y_{k=2}$	$y_{k=3}$
1st row → x_{11}, x_{12}	1	1	0	0
2nd row → x_{21}, x_{22}	3	0	0	1
3rd row → x_{31}, x_{32}	2	0	1	0

~~$y_{(i,k)}$~~

→ $y_1^{1\text{-row}}$ (1 → class) $\log \hat{y}_1^1 + y_2^1 \log \hat{y}_2^1 + y_3^1 \log \hat{y}_3^1$ (2 → class)

→ $y_k^{2\text{-row}}$ $\log y_1^2 \log \hat{y}_1^2 + y_2^2 \log \hat{y}_2^2 + y_3^2 \log \hat{y}_3^2$

→ $y_1^3 \log \hat{y}_1^3 + y_2^3 \log \hat{y}_2^3 + y_3^3 \log \hat{y}_3^3$

now cost fun becomes $CF = y_1^{(1)} \log \hat{y}_1^{(1)} + y_2^{2} \log \hat{y}_2^{(2)} + y_3^{3} \log \hat{y}_3^{3}$

now we have $\hat{y}_1^{1} + \hat{y}_2^{2} + \hat{y}_3^{3}$

$\theta_0^{(1)}$	$\theta_1^{(1)}$	$\theta_2^{(1)}$
$\theta_0^{(2)}$	$\theta_1^{(2)}$	$\theta_2^{(3)}$
θ_0^{3}	θ_1^{3}	θ_3^{3}

$\hat{y}_1 = \sigma(\theta_0^{(1)} + \theta_1^{(1)} u_{11} + \theta_2^{(1)} u_{12})$

$\hat{y}_2 = \sigma(\theta_0^{(2)} + \theta_1^{(2)}(u_{21}) + \theta_2^{(2)}(u_{22}))$

$\hat{y}_3 = \sigma(\theta_0^{(3)} + \theta_1^{(3)}(u_{31}) + \theta_2^{(3)}(u_{32}))$

gradient descent

$\frac{\partial L}{\partial \theta_0^{(1)}}, \frac{\partial L}{\partial \theta_0^{(2)}}$ ---------- 9 diff.

Convergence $\theta_1^{(1)} = \theta_1^{(1)} - \eta \frac{\partial L}{\partial \theta_1^{(1)}}$

Simultaneously change θ

now cost fun becomes $CF = y_1^{(1)} \log \hat{y}_1^{(1)} + y_2^{2} \log \hat{y}_2^{(2)} + y_3^{3} \log \hat{y}_3^{3}$

now we have $\hat{y}_1^{1} + \hat{y}_2^{2} + \hat{y}_3^{3}$

$\alpha_0^{(1)}$	$\alpha_1^{(1)}$	$\alpha_2^{(1)}$
$\alpha_0^{(2)}$	$\alpha_1^{(2)}$	$\alpha_3^{(3)}$
α_0^{3}	α_1^{3}	α_3^{3}

$\hat{y}_1 = \sigma(\alpha_0^{(1)} + \alpha_1^{(1)} u_{11} + \alpha_2^{(1)} u_{12})$

$\hat{y}_2 = \sigma(\alpha_0^{(2)} + \alpha_1^{(2)}(u_{21}) + \alpha_2^{(2)}(u_{22}))$

$\hat{y}_3 = \sigma(\alpha_0^{(3)} + \alpha_1^{(3)}(u_{31}) + \alpha_3^{(3)}(u_{32}))$

gradient descent

$\frac{\partial L}{\partial \alpha_0^{(1)}}, \frac{\partial L}{\partial \alpha_0^{(2)}}$ ---------- 9 diff.

convergence $\alpha_1^{(1)} = \alpha_1^{(1)} - \eta \frac{\partial L}{\partial \alpha_1^{(1)}}$

Simantaneously change α