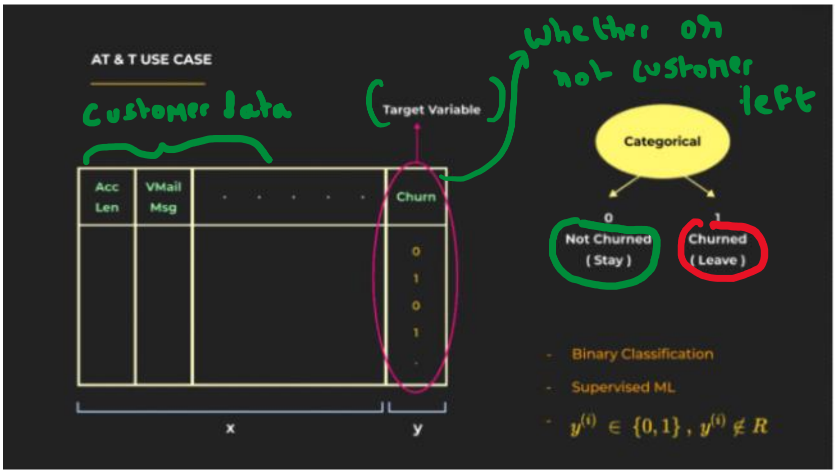
Use Case Introduction

07 August 2025 19:55



Features	Description
state	2-letter code of the US state of customer residence
account_length	Number of months the customer has been with the current telco provider
area_code	string="area_code_AAA" where AAA = 3 digit area code
intl_plan	The customer has international plan
vmail_plan	The customer has voice mail plan
vmail_messages	Number of voice-mail messages
day_mins	Total minutes of day calls
day_calls	Total no of day calls
day_charge	Total charge of day calls
eve_mins	Total minutes of evening calls
eve_calls	Total no of evening calls
eve_charge	Total charge of evening calls

Linear Regression

↳ continous value

{ Sales
price
age }

Logistic Regression

↳ Binary value

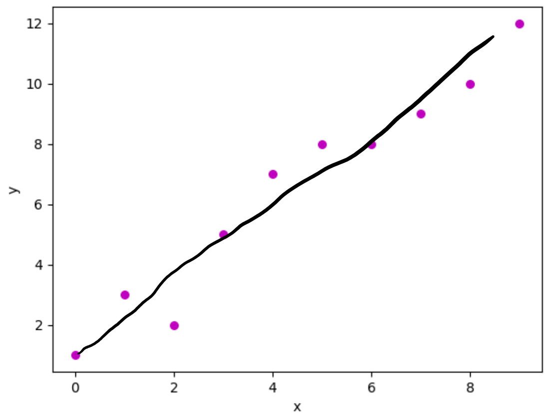
Yes/No
T/F
1/0
} classes

Classification Algorithm

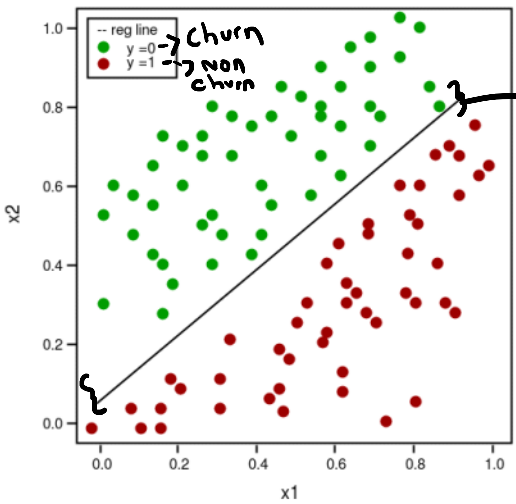
Introduction to Logistic Regression

07 August 2025 19:57

Linear Regression



Logistic Regression



→ This line separates the two classes

Marks (x)	x_1	x_2	x_3	Result	(y_i) → Actual y
0				Fail	0
17				Fail	0
34				Fail	0
51				Pass	1
68				Pass	1
85				Pass	1
100				Pass	1

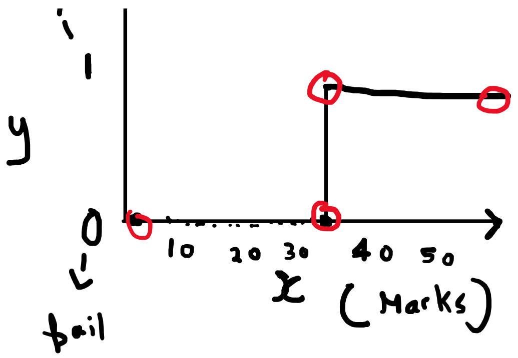
Cut off
⇓
35, 50

We can just Find the right conditions to classify?

Ex :- Marks >= 35 , Pass (y = 1)

Marks < 35, Fail (y = 0)

[→ {step Function}



=> {Step Function}

But this is not differentiable / It is not continuous!

For optimization algo like Gradient Descent to work, functions should be differentiable and continuous!

Sigmoid/ Logistic Function

07 August 2025 20:16

Some transformation of x } transform x into z

📊 Sigmoid-Centered Transformation Table

Marks (x)	$z = g(x)$	Sigmoid(z)	Result
0	-7.0	0.0009	Fail
17	-3.6	0.0266	Fail
34	-0.2	0.4502	Fail
51	3.2	0.9608	Pass
68	6.6	0.9986	Pass
85	10.0	0.99995	Pass
100	13.0	0.999998	Pass

$\rightarrow \frac{1}{1+e^{-(-7)}} = 0.0009$

$\rightarrow \frac{1}{1+e^{-(-3.6)}} = 0.0266$

$\frac{1}{1+e^{-(13)}} = 0.999$

Eulers constant

e is a constant $= 2.718$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$ → Formula for sigmoid

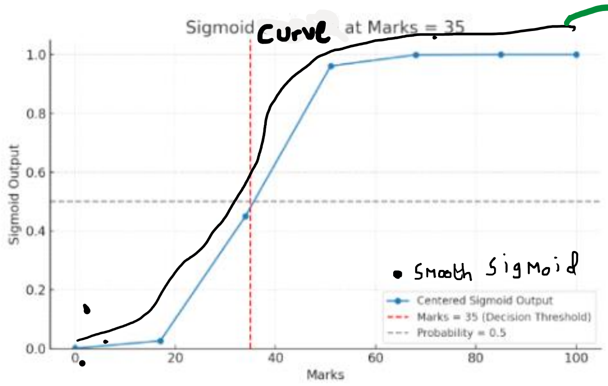
Sigmoid for Z $= \frac{1}{1+e^{-z}}$

Behaviour of sigmoid

Very close to 0 for very negative values

Very close to 1 for very positive values

We interpret sigmoid generated values as probability of 1 (in our case 1 = pass 0 = Fail)

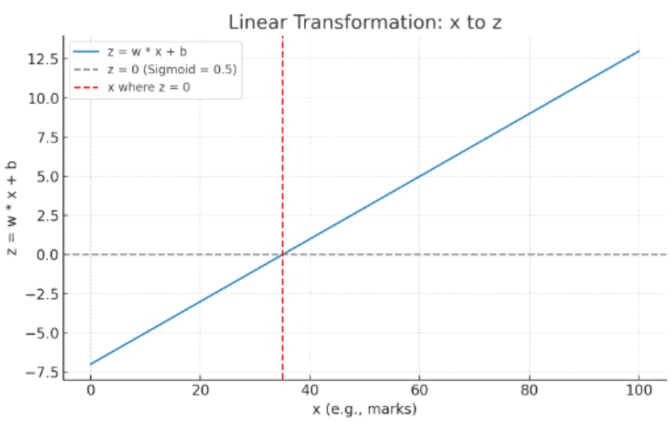


Lot smoother,
continuous
differentiable

But how exactly did we transform x to some value z?

$$\left\{ z = w_1 x + w_0 \;!! \right\}$$

z is a linear transformation of x



But how do we find the right weights w_1 and w_0 for $z = w_1 x + w_0$?

That is where ML comes in!

Just how we find optimal weights in Linear Regression (by minimizing SSE), there is an approach the logistic Regression algorithm follows!

What do we minimize is Linear Regression

$$SSE = \sum(\hat{y} - y_i)^2$$

Quiz

08 August 2025 20:25

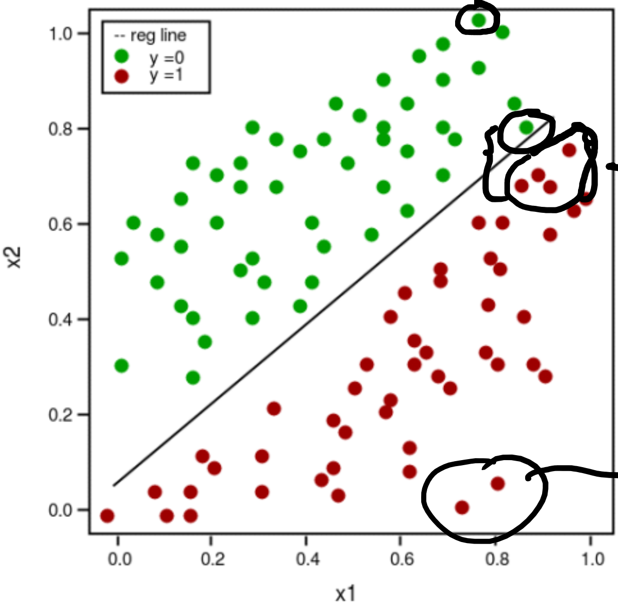
What happens when the input to the sigmoid function is a very large negative value?

Choices

- [ ] The output becomes negative
- [x] The output approaches 0
- [ ] The output approaches 1
- [ ] The output becomes undefined.

Geometrical Representation of sigmoid

08 August 2025 20:22



Probability

⇑

z will be very negative ⇒ sigmoid will be close to 0

z will be close to 0 ⇒ sigmoid will be close to 0.5

z will be very positive ⇒ sigmoid will be close to 1

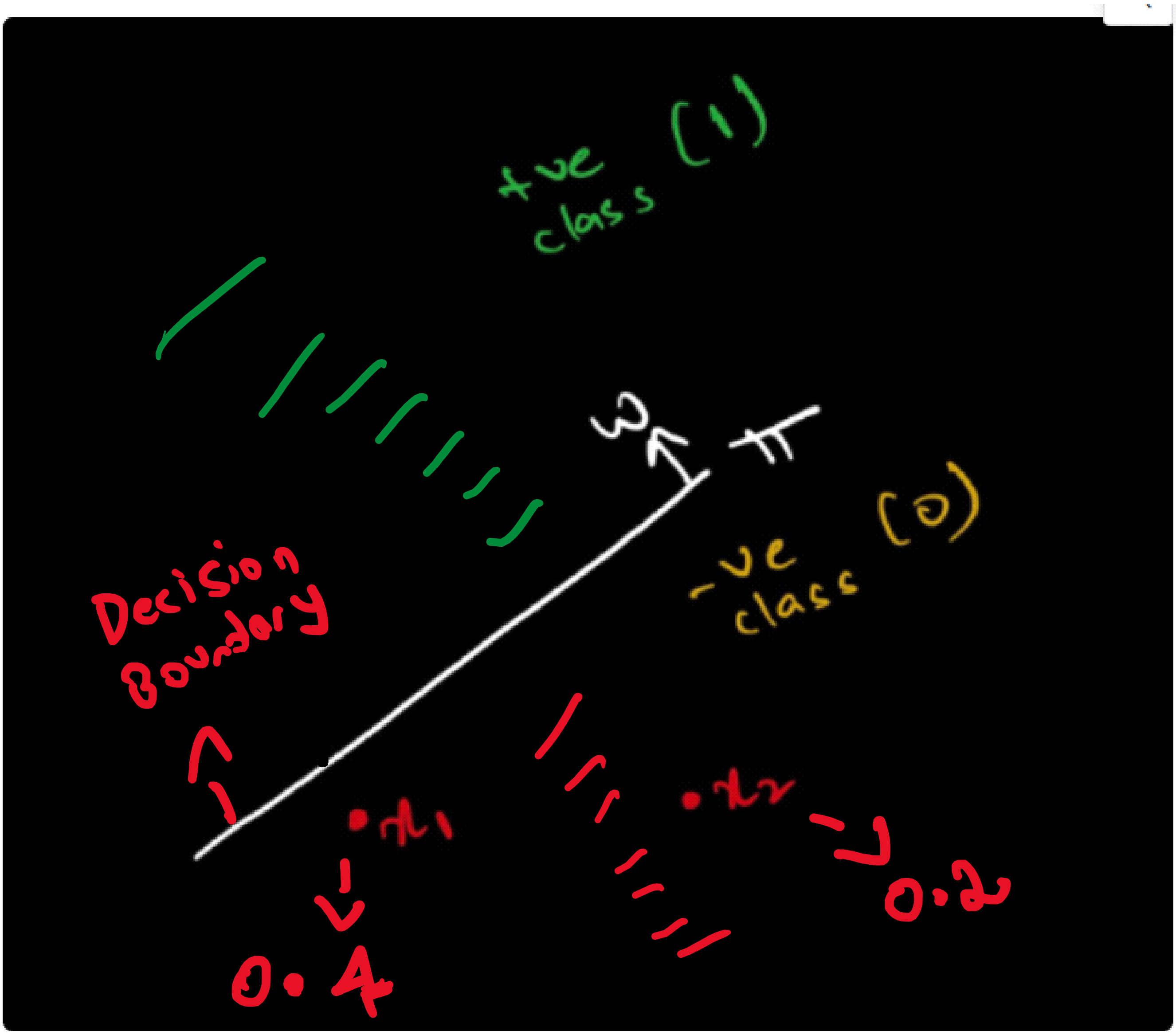
⇓

Probability Close to 1

Points that lie close to the boundary are Uncertain points → probability close to 0.5

Quiz

07 August 2025 20:50



We know that the goal of Logistic Regression is to output a probability (sigmoid).

Let us say we are predicting the probability $\hat{y}$ that someone has diabetes.

$\hat{y}$	y_i
0.8	1
0.6	0
0.4	0

$\hat{y} \to$ Predicted by the Model

$y_i \to$ Actual given by you to the model

In logistic regression, every $\hat{y}$ is associated with actual y called y_i ?

Just like in any other Model.

We derive a formula for something called Likelihood.

$$\left\{ \text{likelihood} = \hat{y}^{\,y_i} \times (1-\hat{y})^{1-y_i} \right\}$$

{Don't get confused !}

Likelihood is just a way to reward our Model if predicted probability is closer to actual.

How? $\hat{y}$ is probability but y_i is always

Example $\to$ if $\hat{y}_i = 0.9$ and $y_i = 1$

Example → if $\hat{y}_i = 0.9$ and $y_i = 1$

$$0.9^1 \times (0.1)^{1-1} = 0.9 \times 0.1^0 = 0.9$$

if $\hat{y} = 0.5$ and $y_i = 1$

$$0.5^1 \times (0.5)^0 = 0.5 \times 1 = 0.5$$

if $\hat{y} = 0.1$ and $y_i = 1$

$$0.1^1 \times 0.9^0 = 0.1 \times 1 = 0.1$$

$\{\hat{y} = 0.1 \quad y_i = 0 \quad 0.1^0 \times 0.9^1 = 0.9\}$

S.No.	y_i	$\hat{y}$	Likelihood
1	1	0.9	0.9 ✓
2	1	0.5	0.5
3	1	0.1	0.1
4	0	0.1	0.9 ✓

It is called likelihood because it tells us how likely the model thinks we will observe that point.

Let me multiply $L_1 \times L_2 \times L_3 \times L_4$

Will the result be higher if all four individual likelihoods are high or low?

By multiplying likelihood of all points

L = likelihood$_1$ × likelihood$_2$ × likelihood$_3$ ..... likelihood$_n$

We will get likelihood of all points

we will get likelihood of all points

or likelihood of our entire data!

Mathematically,

Multiplication Function

$$\text{likelihood of entire data} = \prod_{i=1}^{n} \hat{y}_i^{\,y_i} \times (1-\hat{y}_i)^{1-y_i}$$

Multiply likelihood of all points

or

$$L = \prod_{i=1}^{n} \hat{P}_i^{\,y_i} \times (1-\hat{P}_i)^{1-y_i}$$

Since its harder to differentiate products, We will convert into a {sum.}

Take log on both sides => converts product into sum

$$\log L = \sum_{i=1}^{n} y_i \log \hat{P}_i + (1-y_i) \log (1-\hat{P}_i)$$

{log likelihood}

Goal => We want to Maximize this log likelihood ({because it is a reward!)

But Gradient Descent likes to Minimize things. So, lets convert into a Minimization problem. How? Just add a negative!

Minimize

$$\text{negative log } L = -\sum^{n} y_i \log \hat{P}_i + (1-y_i) \log (1-\hat{P}_i)$$

$$\text{negative } \log L = -\sum_{i=1}^{n} y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)$$

↳ this is called {log loss} or **Cross-Entropy!** } loss function in logistic regression

We want to Minimize this!

Just how we Minimize SSE in Linear Regression

Using

Gradient Descent!

✅ Summary

Concept	Formula	Intuition
[illegible]	[illegible]	[illegible]
Likelihood	$\prod \hat{p}^y (1-\hat{p})^{1-y}$	Match predicted probabilities to actual labels
Log-Likelihood	$\sum y\log(\hat{p}) + (1-y)\log(1-\hat{p})$	Optimized in logistic regression
Objective	Maximize likelihood = minimize binary cross-entropy	Fit confident and accurate predictions

{Maximum Likelihood Estimation} → Gradient Descent

to obtain optimal weights

$w_1 \quad w_0$

$w_1 \quad w_0$

$$z = w_1 x + w_0$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \Big\} \to \text{Probabilities}$$

Quiz

08 August 2025 20:27

Question

Supposedly your y = 0 and ŷ = 0.01 , so what be the log-loss ?

Choices

- [ ] log-loss will be a very high value → log likelihood
- [x] log-loss will be a very low value
- [ ] log-loss will be 0

{ log loss = negative likelihood }

Optimization Process

07 August 2025 23:18

How gradient Descent Minimizes log loss to obtain optimal weights

We first take derivative of 1 point then generalize to m points

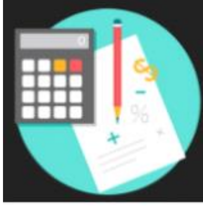
$$L = -[\underbrace{y^{(i)}.log\hat{y}^{(i)}}_{A} + \underbrace{(1-y^{(i)}).log(1-\hat{y}^{(i)})}_{B}]$$

$$\hat{y} = \sigma(w_1x_1 + w_1x_2 + ... + w_jx_j + ...)$$

$$\frac{\partial L_A}{\partial w_j} => \frac{\partial A}{\partial \hat{y}}.\frac{\partial \hat{y}}{\partial z}.\frac{\partial z}{\partial w_j}$$

$$=> \frac{y}{\cancel{\hat{y}}}.\cancel{\hat{y}}(1-\hat{y}).x_j$$

$$=> y(1-\hat{y}).x_j$$



Now, using than rule

$$\frac{\partial L_B}{\partial w_j} = \frac{\partial B}{\partial(1-\hat{y})}.\frac{\partial(1-\hat{y})}{\hat{y}}.\frac{\partial \hat{y}}{\partial z}.\frac{\partial z}{\partial w_j}$$

$$= \frac{1-y}{\cancel{1-\hat{y}}}.(-1).\hat{y}\cancel{(1-\hat{y})}.x_j$$

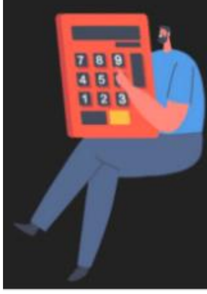
$$= (1-y).\hat{y}.x_j$$

$$\frac{\partial L}{\partial w_j} = \frac{L_A}{\partial w_j} + \frac{\partial L_B}{\partial w_j}$$

$$= y(1-\hat{y})x_j - \hat{y}(1-y)x_j$$

$$= x_j[y - y\hat{y} - \hat{y} + y\hat{y}]$$

$$= [y - \hat{y}].x_j$$



Now, we use the -ve sign we earlier forget

$$=> \frac{\partial L}{\partial w_j} = [\hat{y} - y]x_j$$

Summing it all up

For all pts., i = 1 to m

$$\frac{\partial L}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)}).x_j^{(i)}$$

=) Derivative of loss fn is very similar lin reg

=> This is same as lin. reg.

Diff is

=> This is same as lin. reg.

very similar

Diff is

$$Lin\ Reg => \hat{y} = w^T x + w_0$$

Key Diff

$$Log\ Reg => \hat{y} = \sigma(w^T x + w_0)$$

$$= \frac{1}{1 + e^{-(w^T x + w_0)}}$$

For grad. descent:

$$=> w_j = w_j - \eta \frac{\partial L}{\partial w_j}$$

Quiz

08 August 2025 20:28

Question

In logistic regression, the output of the sigmoid function is interpreted as:

Choices

- [x] Class probabilities
- [ ] Raw scores
- [ ] Error rates
- [ ] Regression coefficients

Accuracy

07 August 2025 23:24

Actual y	Predicted Prob $\hat{p}$	Predicted Class $\hat{y}$	Correct?
1	0.92	1	✅ Yes
0	0.12	0	✅ Yes
1	0.65	1	✅ Yes
0	0.53	1	❌ No
1	0.48	0	❌ No
0	0.06	0	✅ Yes
1	0.85	1	✅ Yes
0	0.34	0	✅ Yes
1	0.29	0	❌ No
0	0.78	1	❌ No

$$\left\{ \text{Accuracy} = \frac{\text{Number of Correct predictions}}{\text{Number of datapoints}} \right\} = \frac{6}{10} = 60\%$$

I will provide a dataset

x_1	x_2	x_3	x_4	x_5	y

Model will learn relationship b/w x and y

x_1, x_2, x_3 etc. will be represented by

$$Z = \underbrace{w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0}$$

How does model figure out weights

Maximum likelihood estimation

Mimize log loss using
gradient descent
to find
optimal weights

We can calculate Z

Pass Z through sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}}$$ where $e = 2.718$
Eulers
constant

Test the model on your testing set

Quiz

08 August 2025 20:28

title: Quiz 5
description:
duration: 60
card_type: quiz_card

Question

What is the main risk of overfitting when tuning hyperparameters in logistic regression?

Choices

- [ ] The model may generalize well to unseen data but poorly on the training data
- [x] The model may perform well on the training data but poorly on unseen data
- [ ] The model may underperform compared to a model with default hyperparameter values
- [ ] The model may be too simple and fail to capture complex relationships in the data

---

Which statement about the step function is true?

Choices

- [ ] It is continuous and differentiable
- [ ] It is continuous but not differentiable
- [x] It is neither continuous nor differentiable
- [ ] It is differentiable but not continuous