A Comprehensive Guide to Logistic Regression

Tanvi Penumudy · Follow
Published in Analytics Vidhya · 8 min read · Jan 14, 2021

'Logistic Regression' is an extremely popular artificial intelligence approach that is used for classification tasks. It is widely adopted in real-life machine learning production settings. We shall get into the depth of this in the next few minutes!

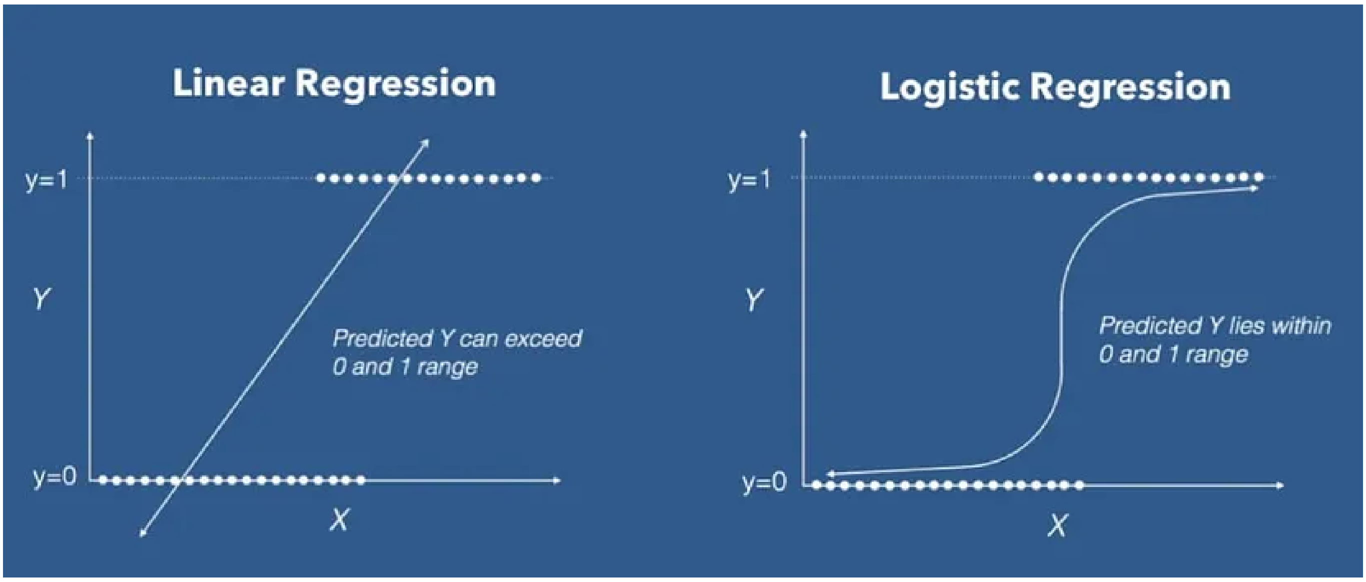


Image Source: Dev.to

Logistic Regression in Layman's Terms

Logistic regression is a machine learning algorithm used to predict the probability that an observation belongs to one of two possible classes.

What does that mean in practice?

We could use the logistic regression algorithm to predict the following:

1. Build an email classifier to tell us whether an incoming email should be marked as ***"spam"*** *or* ***"not spam"***.
2. Check radiological images to predict whether a tumour is ***benign*** or ***malignant.***
3. Pour through historic bank records to predict whether a customer will default on their loan repayments or repay the loan.

How does logistic regression make predictions?

The logistic regression can then be used on novel input data which the model has never seen before (during training).

Let's look at a concrete example:

Imagine that you're tasked to predict whether or not a client of your bank ***will default on their loan repayments.*** The first thing to do is construct a dataset of historic client defaults.

> *The data would contain client demographic information (e.g. age, gender, location, etc.), their financial information (loan size, times that payment was overdue, etc.), and whether they ended up defaulting on a loan or repaying it.*

Client ID	Age	Gender	Loan size	Times payment was overdue	Defaulted?
1	32	F	$10,000	0	No
2	48	M	$30,000	12	Yes
3	51	F	$50,000	1	No

The ***"Yes"*** and ***"No"*** categories can be recoded into ***1*** and ***0*** for the target variable *(computers deal better with numbers than words):*

Client ID	Age	Gender	Loan size	Times payment was overdue	Defaulted?
1	32	F	$10,000	0	0
2	48	M	$30,000	12	1
3	51	F	$50,000	1	0

After this, we would train a ***logistic regression model,*** which would learn a mapping between the *input variables (age, gender, loan size)* and the expected output — to predict the default probability on three new customers:

Client ID	Age	Gender	Loan size	Times payment was overdue	Predicted default
4	31	F	$12,000	1	0.13
5	28	F	$37,000	14	0.98
6	56	F	$48,000	2	0.23

So, what does the new column *Predicted default* tell us?

It states the probability of each of the new customers belonging to class 1 (defaulted on loan). We could come up with a ***threshold value*** (let's say ***0.5***) and anything above that decision threshold could be 1 and the rest 0.

Applications of Logistic Regression

Business Use-Cases

- *Qualify leads*
- *Recommend products*
- *Anticipate rare customer behaviour*

Advantages in Production Deployment

The benefits of logistic regression from an engineering perspective make it more favourable than other, more advanced machine learning algorithms.

- *Ease of use*
- *Interpretability*
- *Scalability*
- *Real-time predictions*

Logistic Regression: Implementation

Logistic regression is a supervised machine learning classification algorithm.

> *For more on this, do check out — [A Beginner's Guide for Getting Started with Machine Learning]()*

Model Representation

Once trained, the model takes the form of a logistic regression equation:

$$y = \frac{1}{1+e^{-(w_o + w_1 x)}}$$

In this equation:

- y is the ***predicted probability*** of belonging to the default class.

> *In binary classification, we mark the default class with 1 and the other class with 0. y states the probability of an example belonging to the default class on a scale from 0 to 1.*

- **1/(1+e^-z)** is the ***sigmoid function.***

- **wo + w1x** is the linear model within logistic regression.

Linear Model

The linear model represents a *linear relationship* between the ***input features*** and the ***predicted output***. The linear part of the entire model can be summarized with the equation:

$$w_o + w_1 x$$

What does each component mean here?

- **x** is the *input variable*. In statistics, x is referred to as an *independent variable*, while machine learning calls it a *feature*.
- ***w0*** is the *bias term*.
- ***w1*** is the *weight* for the *input variable x*.
- In machine learning, we call ***wi*** *weights/parameters* in general.

> *So, why wouldn't we just use the linear model to make predictions about class membership, as we did with linear regression?*

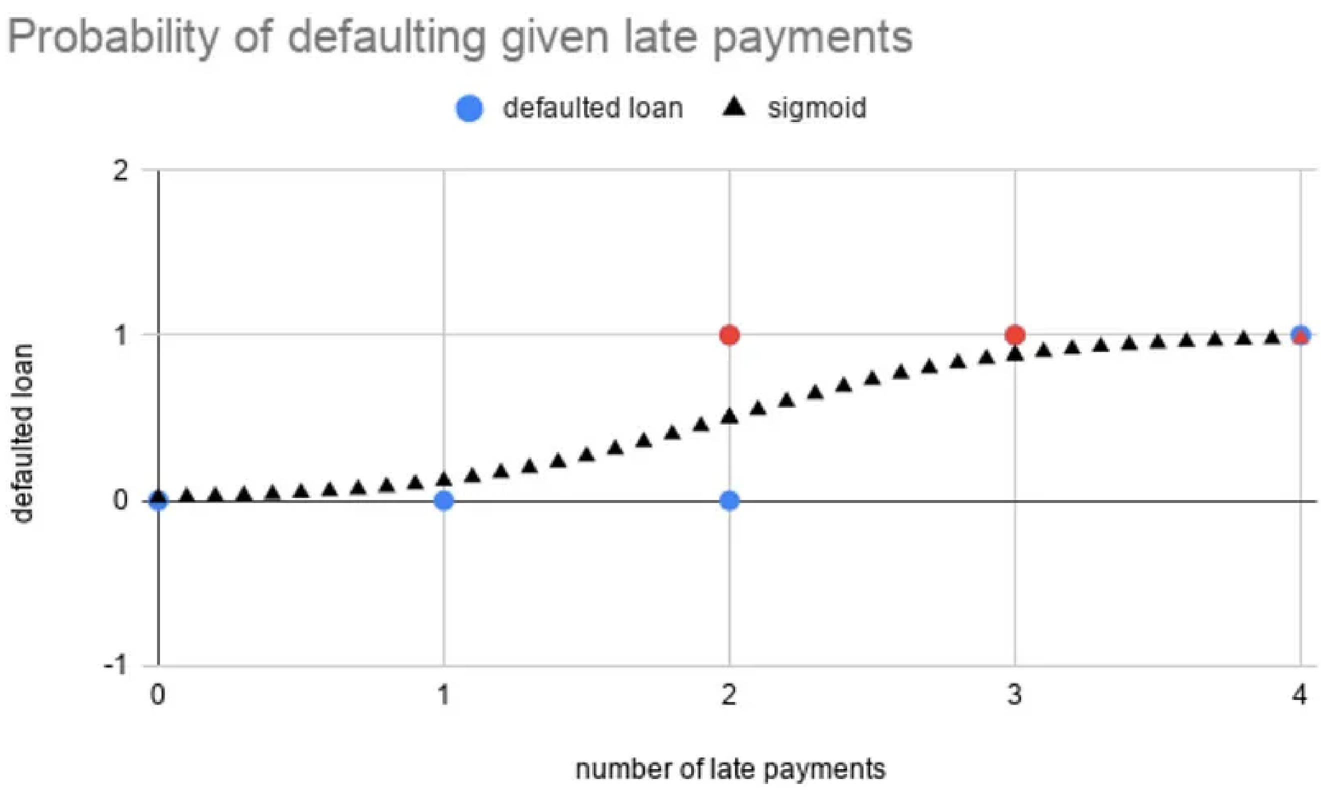
1. *Linear regression predicts probabilities outside of the 0–1 range*
2. For a certain number of late payments (two in this example), it is unclear whether we should categorize them under non-defaulting or defaulting behaviour.

> *A better approach would be to model the probability of default using a **sigmoid function.***

Sigmoid Function

The sigmoid function is a function that produces an ***s-shaped curve.*** It takes any real value as an argument and maps it to a range between 0 and 1.

For the problem above, the sigmoid curve would look like this:



It is used to map the linear model in logistic regression to map the *linear predictions to outcome probabilities (bounded between 0 and 1),* which are easier to interpret for ***class membership.***

> *How do we map class membership probability to predicted class? We need a* ***decision boundary*** *to disambiguate between different probabilities.*

Decision Boundary

A ***decision boundary*** is a threshold that we use to categorize the probabilities of logistic regression into discrete classes.

A decision boundary could take the form:

y = 0 if ***predicted probability*** *< 0.5*

y = 1 if ***predicted probability*** *> 0.5*

Types of Logistic Regression

1. ***Binary logistic regression:*** The target variable takes one of two possible categorical values.
2. ***Multinomial logistic regression:*** The target variable takes one of three or more possible categorical values.
3. ***Ordinal logistic regression:*** This is similar to multiple logistic regression, except the target categorical variables are ordered.

> *Irrespective of the type of logistic regression, training the logistic regression model follows a similar process in all cases.*

Training

The aim of training the logistic regression model is to figure out the ***best weights for our linear model*** within the logistic regression.

> *In machine learning, we compute the optimal weights by optimizing the cost function.*

Cost Function

The ***cost function J(Θ)*** is a formal representation of an objective that the algorithm is trying to achieve.

In the case of logistic regression, the cost function is called LogLoss (or Cross-Entropy) and the goal is to minimize the following cost function equation:

$$J(\theta) =- \frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))\right]$$

You just need to understand the principles behind it:

- The cost function checks what the average error is between actual class membership and predicted class membership.

> *This is caused by the specific selection of weights within our linear model.*

- The cost function not only penalizes big errors but also errors which are too confident (too close to 0 or 1).

> *This guarantees that our predictions stay within the 0–1 range, exclusive.*

So, how do we achieve a low value for our cost function *(aka, a model with good predictions)*? We use ***gradient descent.***

Gradient Descent

Gradient descent is a method of changing weights based on the loss function for each data point. We calculate the ***LogLoss*** cost function at *each input-output data point.*

- We take a partial derivative of the weight and bias to get the slope of the cost function at each point.
- Based on the slope, gradient descent updates the values for the bias and the set of weights, then re-iterates the training loop over new values.
- This iterative approach is repeated until a minimum error is reached, and gradient descent cannot minimize the cost function any further.
- We can change the speed at which we reach the optimal minimum by adjusting the learning rate.

> *A high learning rate changes the weights more drastically, while a low learning rate changes them more slowly.*
>
> *There is a trade-off in the size of the learning rate. Too low, and you might be waiting forever for your model to converge on the best set of weights; too high, and you risk missing the best set of weights because the model would not converge.*

Model Evaluation

There are two main metrics for evaluating how well our model functions after we've trained it:

- ***Accuracy:*** Represents the percentage of correctly classified samples.

> *An accuracy score of 90% would tell us that our logistic regression model correctly classified 90% of all examples.*

- ***ROC AUC:*** Area Under the *Receiver Operating Characteristic Curve (ROC AUC)* describes the relationship between the *true positive rate (TRP)* — that is, the ratio of samples that we correctly predicted belonging to the correct class — versus the *false positive rate (FPR)* — that is, the ratio of samples for which we incorrectly predicted their class membership.

> *ROC AUC is preferable to accuracy, especially in multiclass prediction settings or when we have a class imbalance problem.*

Improving our Model

There are multiple methods to improve your Logistic Regression model.

There are a few techniques (in preprocessing) that are employed for model improvement in Linear Regression as elaborated in — ***Everything You Need to Know About Linear Regression***

Logistic regression has additional assumptions and needs for cleaning:

1. ***Binary output variable:*** Transform your output variable into 0 or 1.
2. ***Failure to converge:*** The *maximum likelihood estimation* model (the 'maths') behind logistic regression assumes that no single variable will perfectly predict class membership. In the event that you have a feature that perfectly predicts the target class, the algorithm will try to assign it infinite weights (because it is so important) and thus will fail to converge to a solution.