Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for binary classification tasks. It is used for predicting the categorical dependent variable using a given set of independent variables. It is used to predict the probability that an instance belongs to a particular class, typically denoted as class 1, given its features. The model is based on the logistic function (sigmoid function), which maps any real-valued number to a value between 0 and 1. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.

## Sigmoid function

For Logistic Regression, we can start by using our linear regression model, *𝑓***𝐰**,(**𝐱**(*𝑖*))=**𝐰**⋅**𝐱**(*𝑖*)+*𝑏*, to predict y given x

However, we would like the predictions of our classification model to be between 0 and 1 since our output variable y is either 0 or 1. This can be accomplished by using a "sigmoid function" which maps all input values to values between 0 and 1.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1.
* The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the sigmoid function or the logistic function.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

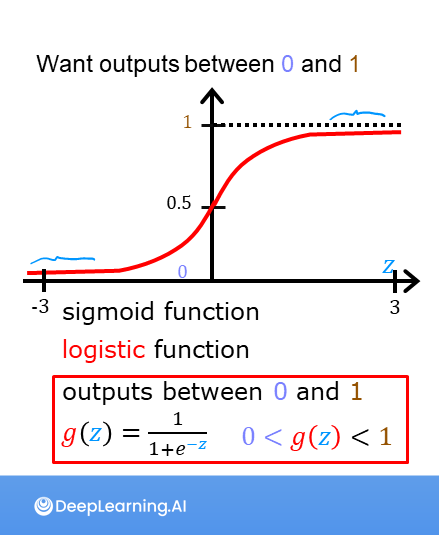
The formula for a sigmoid function is as follows -

g(𝑧)=1/1+e^-z

In the case of logistic regression, z (the input to the sigmoid function), is the output of a linear regression model.

In the case of a single example, z is scalar.

In the case of multiple examples, z may be a vector consisting of m values, one for each example.



Assumptions for Logistic Regression**:**

* The dependent variable must be categorical in nature.
* The independent variable should not have multi-co linearity.

# Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

* We know the equation of the straight line can be written as:

Logistic Regression in Machine Learning

* In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

Logistic Regression in Machine Learning

* But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

Logistic Regression in Machine Learning

# Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

* **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
* **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
* **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Steps in Logistic Regression:** To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

* Data Pre-processing step
* Fitting Logistic Regression to the Training set
* Predicting the test result
* Test accuracy of the result(Creation of Confusion matrix)
* Visualizing the test set result.

**1. Data Pre-processing step:** In this step, we will pre-process/prepare the data so that we can use it in our code efficiently. By executing the above lines of code, we will get the dataset as the output. Now we will split the dataset into a training set and test set. In logistic regression, we will do feature scaling because we want accurate result of predictions. Here we will only scale the independent variable because dependent variable has only 0 and 1 values.

**2. Fitting Logistic Regression to the Training set:**

We have well prepared our dataset, and now we will train the dataset using the training set.

**3. Predicting the Test Result**

Once the model is well trained on the training set, we predict the result by using test set data.

**4. Test Accuracy of the result: We check the actual and predicted values by Logistic Regression.**

**5. Visualizing the training set result**

Finally, we will visualize the training set result.

**Visualizing the test set result:**

Our model is well trained using the training dataset. Now, we will visualize the result for new observations (Test set).

Hence our model is pretty good and ready to make new predictions for this classification problem.

A logistic regression model applies the sigmoid to the familiar linear regression model as shown below:

f**𝐰**,(**𝐱**(𝑖))=g(**𝐰**⋅**𝐱**(𝑖)+b)

and **𝐰**⋅**𝐱** is the vector dot product:

**𝐰**⋅**𝐱**=*𝑤*0*𝑥*0+*𝑤*1*𝑥*1

We interpret the output of the model (*𝑓***𝐰**,(*𝑥*)) as the probability that *𝑦*=1 given **𝐱** and parameterized by **𝐰** and *𝑏*. Therefore, to get a final prediction (*𝑦*=0or *𝑦*=1) from the logistic regression model, we can use the following heuristic -

if *𝑓***𝐰**,*𝑏*(*𝑥*)>=0.5, predict *𝑦*=1

if *𝑓***𝐰**,*𝑏*(*𝑥*)<0.5, predict *𝑦*=0

For a logistic regression model, *𝑧*=**𝐰**⋅**𝐱**+*𝑏*. Therefore,

if **𝐰**⋅**𝐱**+*𝑏*>=0, the model predicts *𝑦*=1

if **𝐰**⋅**𝐱**+*𝑏*<0, the model predicts *𝑦*=0

A surface plot of the cost using a *squared error cost*:

*𝐽*(*𝑤*,*𝑏*)=1/2*𝑚*∑*𝑖*=0 to*𝑚*−1(*𝑓𝑤*,*𝑏*(*𝑥*(*𝑖*))−*𝑦*(*𝑖*))^2

where ,

*𝑓𝑤*,*𝑏*(*𝑥*(*𝑖*))=*𝑠𝑖𝑔𝑚𝑜𝑖𝑑* (*𝑤𝑥*(*𝑖*)+*𝑏*)

While this produces a pretty interesting plot, the surface above not nearly as smooth as the 'soup bowl' from linear regression!

Logistic regression requires a cost function more suitable to its non-linear nature. This starts with a Loss function

Logistic Regression uses a loss function more suited to the task of categorization where the target is 0 or 1 rather than any number.

**Definition: Loss** is a measure of the difference of a single example to its target value while the  
**Cost** is a measure of the losses over the training set

*𝑙𝑜𝑠*(*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*)),*𝑦*(*𝑖*)) is the cost for a single data point, which is:

*𝑙𝑜𝑠𝑠*(*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*)),*𝑦*(*𝑖*))=

−log(*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*))) if *𝑦*(*𝑖*)=1

−log(1−*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*))) if *𝑦*(*𝑖*)=0

*𝑓***𝐰**,(**𝐱**(*𝑖*)) is the model's prediction, while *𝑦*(*𝑖*) is the target value.

*𝑓***𝐰**,(**𝐱**(*𝑖*))=*𝑔*(**𝐰**⋅**𝐱**(*𝑖*)+*𝑏*) where function *𝑔* is the sigmoid function.

The defining feature of this loss function is the fact that it uses two separate curves. One for the case when the target is zero or (*𝑦*=0) and another for when the target is one (*𝑦*=1). Combined, these curves provide the behavior useful for a loss function, namely, being zero when the prediction matches the target and rapidly increasing in value as the prediction differs from the target.

# Logistic Gradient Descent

Recall the gradient descent algorithm utilizes the gradient calculation:

repeat until convergence:

{*𝑤𝑗*=*𝑤𝑗*−*𝛼* ∂(**𝐰**,*𝑏*)/∂*𝑤𝑗* for j := 0..n-1

*𝑏*=*𝑏*−*𝛼*∂*𝐽*(**𝐰**,*𝑏*)∂*𝑏*

}

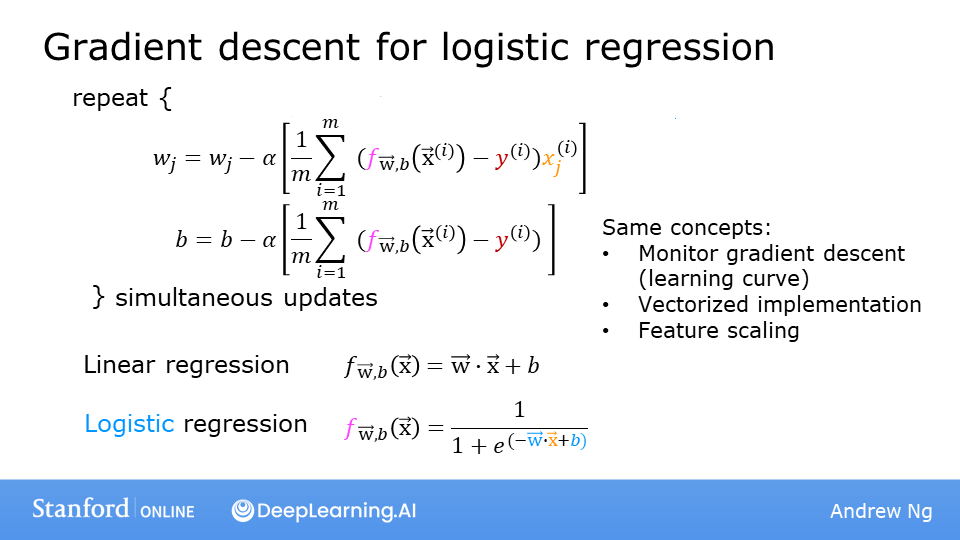
Where each iteration performs simultaneous updates on *𝑤𝑗* for all *𝑗*, where

∂*𝐽*(**𝐰**,*𝑏*)/∂*𝑤𝑗*=1/*𝑚*∑*𝑖*=0 to*𝑚*−1(*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*))−*𝑦*(*𝑖*))*𝑥*(*𝑖*)*𝑗*

∂(**𝐰**,*𝑏*)/∂*𝑏* =1*𝑚*∑*𝑖*=0 to *𝑚*−1(*𝑓***𝐰**,*𝑏*(**𝐱**(*𝑖*))−*𝑦*(*𝑖*))

* m is the number of training examples in the data set
* *𝑓***𝐰**,*𝑏*(*𝑥*(*𝑖*)) is the model's prediction, while *𝑦*(*𝑖*) is the target

 For a logistic regression model  
*𝑧*=**𝐰**⋅**𝐱**+*𝑏*  
*𝑓***𝐰**,*𝑏*(*𝑥*)=*𝑔*(*𝑧*)  
where *𝑔*(*𝑧*) is the sigmoid function:  
*𝑔*(*𝑧*)=11+*𝑒*−*𝑧*



Logistic Regression is a fundamental and widely used binary classification model that finds its applications in various fields, including machine learning, statistics, and data science. Its popularity stems from its simplicity, interpretability, and effectiveness in scenarios where the target variable is binary.

Throughout this detailed explanation, we have explored the underlying principles and mechanics of the Logistic Regression model. By making some crucial assumptions about the linearity of the log-odds and the independence of errors, the model is able to relate the features to the probability of an instance belonging to the positive class, which is essential for binary classification tasks.

The logistic function, or sigmoid function, plays a pivotal role in Logistic Regression. This transformation function maps the linear combination of features and their respective coefficients to a probability value between 0 and 1. By adopting the odds ratio and the logit (log-odds) concepts, we effectively model the relationship between the features and the probability of the positive class in a linear fashion, making it feasible to apply standard linear regression techniques.

The heart of the Logistic Regression model lies in the process of fitting it to the data. The ultimate objective is to determine the optimal coefficients that maximize the likelihood of the observed data, given the model's assumptions. This process is usually accomplished through Maximum Likelihood Estimation (MLE) or, equivalently, minimizing the negative log-likelihood function. Finding these optimal coefficients is often a complex task since there is no closed-form solution. Nevertheless, numerical optimization algorithms, such as Gradient Descent or Newton's Method, step in to iteratively refine the coefficients to their optimal values.

The predictive capability of the Logistic Regression model is highly dependent on the learned coefficients. Once the model is trained, it can make predictions for new instances by calculating the log-odds of the positive class using the learned coefficients and the features of the new data point. Subsequently, the logistic function is applied to transform the log-odds into a probability value, which effectively quantifies the likelihood of the instance belonging to the positive class. By applying a decision threshold, often set to 0.5, the model classifies instances into either the positive class or the negative class, completing the binary classification process.

In summary, Logistic Regression is a powerful tool in the data scientist's toolkit, offering valuable insights into the relationships between features and the probability of a binary outcome. Its simplicity and interpretability make it an attractive choice for scenarios where understanding the impact of individual features on the outcome is as crucial as making accurate predictions. However, it is important to note that Logistic Regression is limited to binary classification tasks and may not be suitable for more complex multi-class problems. As such, while appreciating the strengths of Logistic Regression, data scientists must always carefully consider the nature of their data and the requirements of the classification problem at hand to choose the most appropriate model for their specific use case.