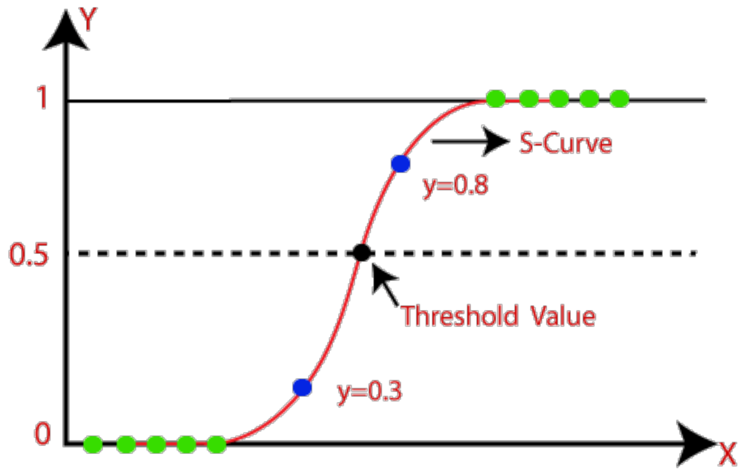
Logistic Regression

Logistic Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- In Logistic regression, instead of fitting a regression line, we fit an **"S" shaped logistic function**, which predicts 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.



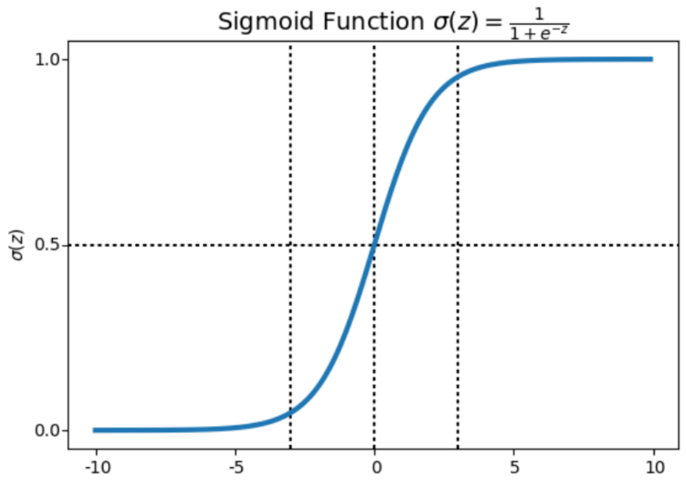
Sigmoid function

- In logistic regression our target variable is categorical but predictor can be numerical or categorical.
- Logistic regression returns the probability of each observation being classified as positive event (probability of default in loan)
- As inputs can also be continuous their combination can range from **-∞ to ∞** but we need the output in form of **probability** which **ranges from 0 to 1** therefore we need a function which can do this conversion.

Sigmoid function

The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

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



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

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

Cost Function in Logistic Regression

- In linear regression we have seen that to measure the error in your predictions we use different error metrics which are also called as cost function (MSE,MAE).
- Similarly in logistic regression also we have a cost function which helps us to assess the predictions made by the algorithm.

Logistic regression Loss and cost function

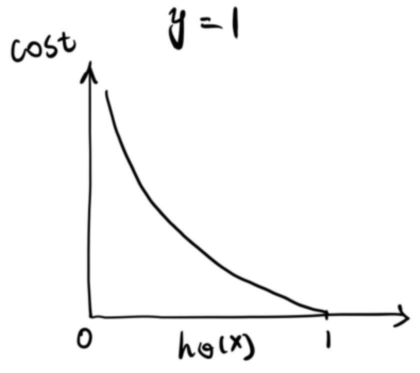
Loss Function for Logistic Regression is $-y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - (1-y^{(i)})\log\left(1-h_\theta\left(x^{(i)}\right)\right)$

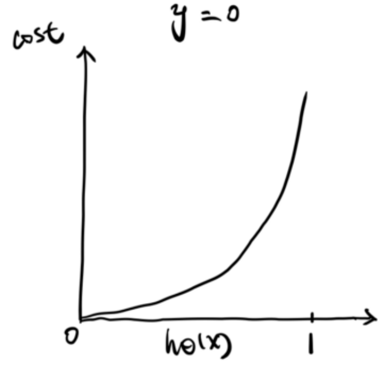
And Cost function is $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}[-y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - (1-y^{(i)})\log\left(1-h_\theta\left(x^{(i)}\right)\right)]$ $m = number\ of\ samples$

$y^{(i)}$ is actual value of target variable (0 or 1)

$h_\theta(x^{(i)})$ is the predicted probability of positive category

Cases	Actual value	Predicted value	Loss
Case 1	0	0	-log (1-$h_\theta(x^{(i)})$) ≈ 0
Case 2	0	1	-log (1-$h_\theta(x^{(i)})$)
Case 3	1	0	-log ($h_\theta(x^{(i)})$)
Case 4	1	1	-log ($h_\theta(x^{(i)})$) ≈ 0





Logistic Regression equation

- In linear regression we have seen that the equation takes the following form

$$\mathrm{Y} = b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

- But in logistic regression our aim is to get the probability of observation being classified as positive class. So the equation in logistic regression takes the following form

$$\log(\frac{p}{1-p}) = b_1x_1 + b_2x_2 + \cdots + b_nx_n \quad \text{or} \quad \mathrm{Z} = b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

Z = $\log(\frac{p}{1-p})$ is called as log odds of probability of success

Therefore in logistic regression we are trying to model log odds of success from input variables.

Conversion of log odds to probability

$\log(\frac{p}{1-p})$ = Z

p= e^Z(1-p)

p = e^Z $-$p e^Z

p(1+ e^Z) = e^Z

p = $\frac{e^Z}{(1+ e^Z)}$ = $\frac{1}{(1+ e^{-Z})}$

Gradient Descent Algorithm

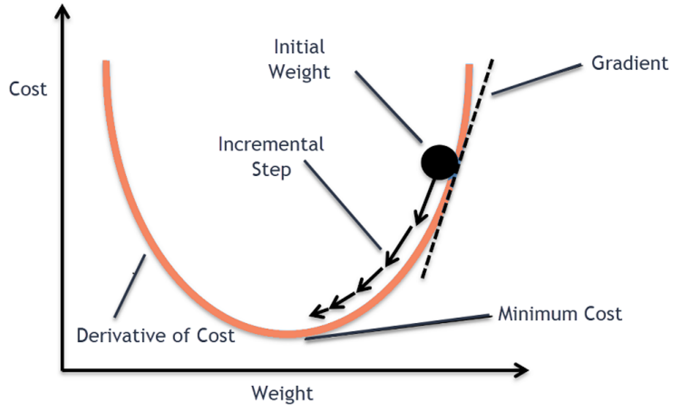
- Just like in linear regression we use method of ordinary least squares (OLS) to determine the best values of coefficients ,Here in logistic regression we use **Gradient descent** algorithm to identify the best values of the coefficients of log odds equation.

Gradient Descent

Gradient descent is an iterative optimization algorithm for finding the local minimum of a function.

To find the local minimum of a function using gradient descent, we must take **steps proportional to the negative** of the gradient **(move away from the gradient)** of the function at the current point.

If we take **steps proportional to the positive** of the gradient **(moving towards the gradient), we will approach a local maximum of the function**, and the procedure is called Gradient Ascent.



Gradient Descent Algorithm

- The goal of the gradient descent algorithm is to minimize the given function (say cost function). To achieve this goal, it performs two steps iteratively:

1.Compute the gradient (slope), the first order derivative of the function at that point.
2.Make a step (move) in the direction opposite to the gradient, opposite direction of slope increase from the current point by alpha times the gradient at that point.

Gradient descent algorithm

```
repeat until convergence {
    $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$
        (for $j = 1$ and $j = 0$)
}
```

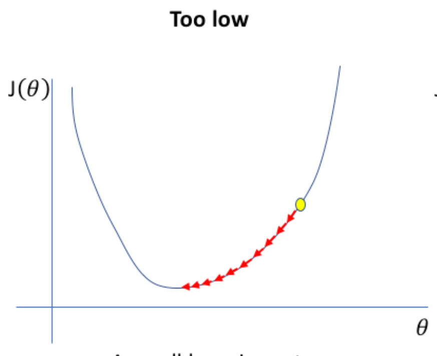
Gradient Descent Algorithm

Alpha – The Learning Rate

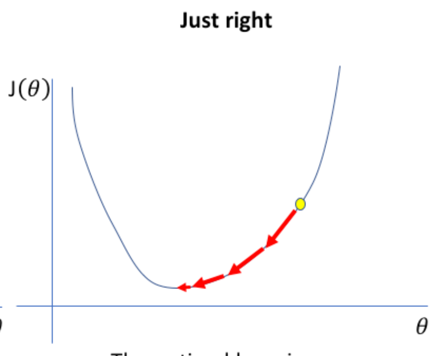
We have the direction we want to move in, now we must decide the size of the step we must take.

It must be chosen carefully to end up with local minima.
•If the learning rate is too high, we might OVERSHOOT the minima and keep bouncing, without reaching the minima
•If the learning rate is too small, the training might turn out to be too long

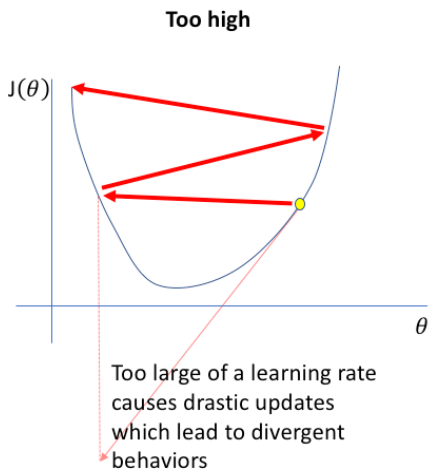
a) Learning rate is optimal, model converges to the minimum
b) Learning rate is too small, it takes more time but converges to the minimum
c) Learning rate is higher than the optimal value, it overshoots but converges
d) Learning rate is very large, it overshoots and diverges, moves away from the minima, performance decreases on learning



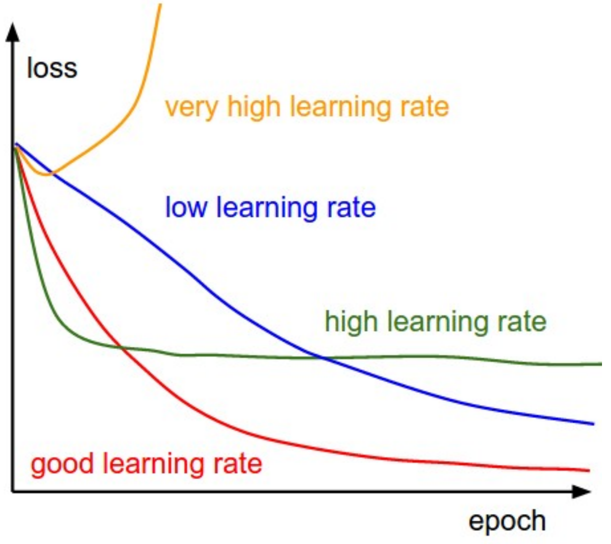
A small learning rate requires many updates before reaching the minimum point



The optimal learning rate swiftly reaches the minimum point



Too large of a learning rate causes drastic updates which lead to divergent behaviors



Logistic Regression

Assumptions of logistic regression

Since Logistic regression is a parametric model it has some assumptions which are as follows:

1) The target variable should be categorical.
2) The independent features should not be correlated .

   This assumption can be checked with the help of VIF and we can use chi squared test for categorical variables.

Parameters of logistic regression in sklearn:

penalty : ***{'l1', 'l2', 'elasticnet', 'none'}***

'none': no penalty is added;
'l2': add a L2 penalty term and it is the default choice;
'l1': add a L1 penalty term;
'elasticnet': both L1 and L2 penalty terms are added.

c ***:float, default=1.0***

Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization

solver ***:{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'***