Re-look

Binary logistic regression

1/[1+e- (w0+w1a1+ w2b1+)]

Suppose we have n-input data set as input vector [X]

|  |  |  |
| --- | --- | --- |
| X= |  | (Sigmoid function) |



|  |  |  |  |
| --- | --- | --- | --- |
| = |  |  | Cutoff 0.5 |

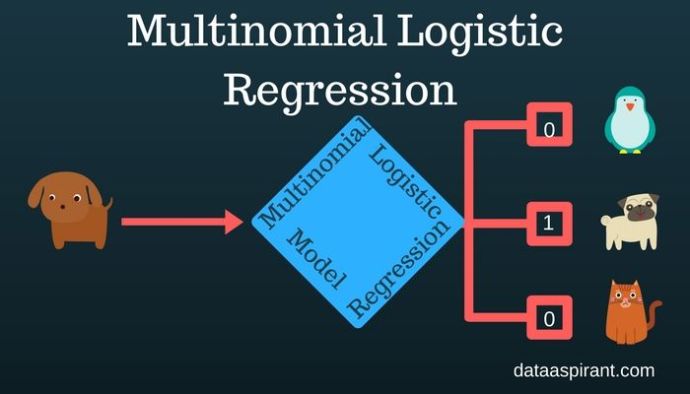
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | X | Sigmoid |  |  | Cutoff 0.5 |

XW+w0

Two Types of MultiClass Classification

• Multi-label Classification - each instance can be assigned more than one labels

• Multinominal Classification - each instance appears in exactly one class (classes are exclusive)



* **Sigmoid function:** used in the logistic regression model for binary classification.
* **Softmax function:** used in the logistic regression model for multiclassification.

(<https://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/>)

|  |  |
| --- | --- |
| Sigmoid Function | Softmax Function |

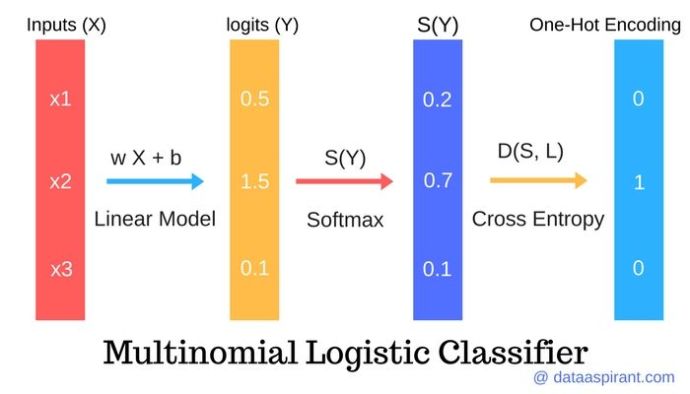
**Difference Between Sigmoid Function and Softmax Function**

The below are the tabular differences between Sigmoid and Softmax function.

|  |  |  |
| --- | --- | --- |
|  | **Softmax Function** | **Sigmoid Function** |
| 1 | Used for multi-classification in logistic regression model. | Used for binary classification in logistic regression model. |
| 2 | The probabilities sum will be 1 | The probabilities sum need not be 1. |
| 3 | Used in the different layers of neural networks. | Used as activation function while building neural networks. |
| 4 | The high value will have the higher probability than other values. | The high value will have the high probability but not the higher probability. |

Multinominal Classification

The underline technique will be same like the [logistic regression for binary classification](https://dataaspirant.com/2017/03/02/how-logistic-regression-model-works/) until calculating the probabilities for each target. Once the probabilities were calculated. We need to transfer them into **one hot encoding** and uses the cross entropy methods in the training process for calculating the properly optimized weights.



Multi-class in practice:

– one weight vector(W) for each class

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | Sigmoid | Y= |

Now apply Softmax function

S(Y)

S(Y1)= Y1/[ Y1+ Y2+ Yn]

S(Y2)= Y2/[ Y1+ Y2+ Yn]

S(Yn)= Yn/[ Y1+ Y2+ Yn]



Now apply cross entropy

(Implementation of one-vs-all)

The Cross-entropy is a [**distance calculation function**](https://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/) which takes the calculated probabilities from softmax function and the created one-hot-encoding matrix to calculate the distance. For the right target class, the distance value will be less, and the distance values will be larger for the wrong target class.

Loss function

(converge the training process/adjust the weight)

The input parameters for the loss function is the calculated weights and all the training observations. The function calculates the distance between the predicted class using the calculated weights for all the features in the training observation and the actual target class.

If the loss function value is fewer means with the estimated weights, we are confident to predict the target classes for the new observations (From test set).  In the case of high loss function value, the process of calculating the weights will start again with derivated weights of the previously calculated weights.

The process will continue until the **loss function value is less.**