Content

1. What is Logistic Regression?
2. How does it work?
3. How to implement Logistic Regression in Python?
4. How to tune Parameters of LR?
5. Pros and Cons associated with LR

**What is Logistic Regression**

Logistic regression is one of the most popular supervised classification algorithm. This classification algorithm mostly used for solving binary classification problems.  Logistic regression algorithm can also use to solve the multi-classification problems.

**“*Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function” (Wikipedia)***

The target variable (Output) can take only discrete values such as True or False, Yes or No, etc. for given set of features(Inputs).

Based on the number of categories, Logistic regression can be classified as:

1. **binomial:** Target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.

Examples**:** Email -> spam or not, Using weather information predicting-> rain or no rain, Based on customer profile, bank predict to give loan or not.

1. **multinomial:** Target variable can have more than 2 possible types which are not ordered like “disease A” vs “disease B” vs “disease C”.  
   Examples: Identifying the different kinds of vehicles.
2. **ordinal:** It deals with target variables with ordered categories. For example, a test score can be categorized as: “very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

**How does it work**

Step 1. Logistic regression model will take the feature values and calculates the probabilities using the **sigmoid** or **softmax** functions.

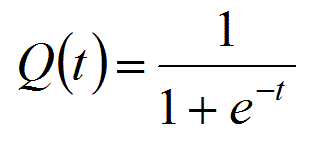
Sigmoid function -> used for binary classification problems   
Softmax function -> used of multi-classification problems.

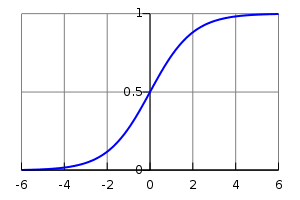
Step 2. Later the calculated probabilities used to find the target class.

Step 3. In general, the high probability class treated as the final target class.

**Sigmoid Function:** Also known as logistic function

The logistic function/sigmoid function, takes any real input (X1, X2…. Xn) and outputs a value between 0 and 1.  
The logistic function is defined as follows:





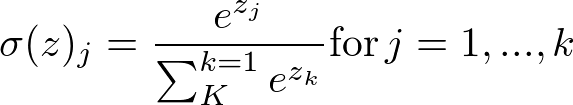
We can infer from above graph that:

* Q(t) tends towards 1 as t -> + infinity
* Q(t) tends towards 0 as t -> - infinity
* Q(t) is always bounded between 0 and 1

**Softmax Function:** A popular function to calculate the **probabilities** of the events.

It returns the high probability value for the high scores and low probabilities for the low scores.

* The calculated probabilities will be in the range of 0 to 1.
* The sum of all the probabilities is equals to 1.

The Softmax function is defined as follows:  
 

Numerator the e-power to the values of the Logit   
Denominator calculates the sum of the e-power to the values of all the Logits.

This function takes each value (Logits) and find the probability.

import numpy as np

def softmax(scores):

    """

    Calculate the softmax for the given scores

    :param scores:

    :return:

    """

    return np.exp(scores) / np.sum(np.exp(scores), axis=0)

scores = [8, 5, 2]

if \_\_name\_\_ == "\_\_main\_\_":

    logits = [8, 5, 2]

    print "Softmax :: ", softmax(logits)