**Logistic Regression and Confusion Matrix**

**Introduction:**

Logistic regression is one of the fundamental algorithms in classification Machine Learning algorithms. It is important to understand how logistic algorithm works and what are the loss functions used to build the best model for classification. These concepts play important role in understanding the other classification algorithms in machine learning.

**Why logistic algorithm is linear?:**

In classification problem, usually the output of an algorithm is probability of a data point belonging to a certain class. This can be achieved using a sigmoidal function, which is given below:

Eq. 1

In logistic regression sigmoidal function is used and it gives the probability as continuous value which is between 0 and 1. In *Eq. 1*, term *ax+b* is a linear term and it is the same equation the one used in linear regression to build the relation between dependent and independent variables. By applying few mathematical operations to *Eq. 1*, one can show that:

Eq. 2

Eq. 3

Eq. 4

Eq.5

Taking log at both sides:

Eq. 6

Left hand side of the Eq.6 is log(odds). Odds is the ratio of probability of an event occurring to probability of opposing event occurring. In logistic regression, event is classification. It is clear from Eq.6 that the log(odds) has linear relation with the independent variable, which is similar to linear regression model. By replacing *ax+b* with log(odds), Eq.1 will result in:

Eq. 7

In logistic regression, linear model (line, surface or hyper surface) is built to predict log(odds) using independent variables. The predicted log(odds) is used to predict the probability of a datapoint belonging to a certain class using Eq.7. This probability prediction is put through a loss function to optimize the model.

Since the logistic classification uses linear regression technique to predict the probability, the algorithm is classified under linear regression.

**Loss Function:**

The loss function used in classification generally is the cross-entropy loss function. There are two cross-entropy loss functions:

1. Binary classification loss function

Binary classification refers to assigning the object/datapoint to one of the two classes. The loss function used is binary cross entropy loss function and it is given below:

Where,

*y* = is the true probability of data point *x*;

*p(x)* = is the predicted probability of data point *x*;

1. Multi-class classification loss function

Multi-class classification loss function is used when the number of classes to assign are more than 2. It is a generalization of the binary cross entropy loss function. The loss function is given by:

Where,

*yi* = is one hot target vector;

*yij* =

*pij(xi)* = is the predicted probability of data point *xi* belonging to class j;

**Confusion matrix:**

Confusion matrix helps in evaluating the performance of the classifier by providing the summary of prediction and ground reality. The general form of confusion matrix for binary classification is given in Table 1. However, the confusion matrix from sklearn.metrics is different and it is given in table Table 2. To support my claim, I am presenting the confusion matrix obtained using sklearn.metrics library in Figure 1. This has been obtained for the diabetic prediction problem, and dataset and .ipynb files are provided in github repository <https://github.com/phegde127/Logistic_Regression>. In this classification problem, 0 represents non-diabetic class and 1 represents diabetic class.

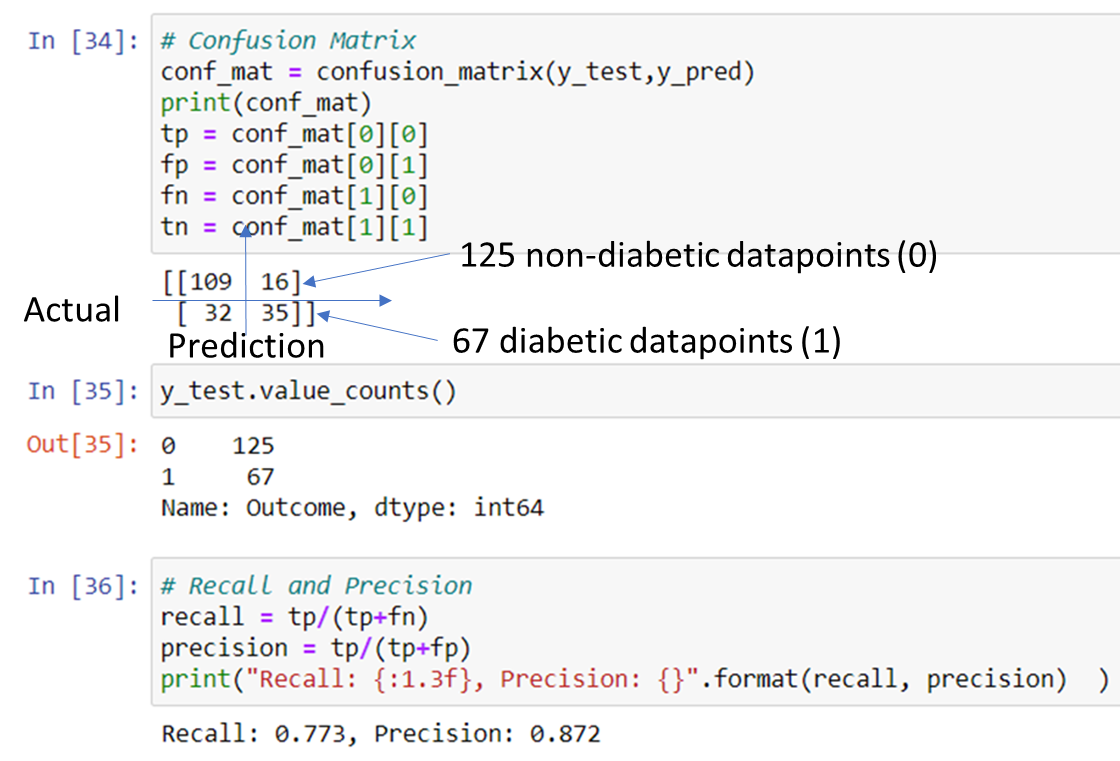
|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual/Ground Reality | | |
| Prediction |  | True | False |
| True | True Positive (TP) | False Positive (FP) |
| False | False Negative (FN) | True Negative (TN) |

**Table 1: General form of confusion matrix**

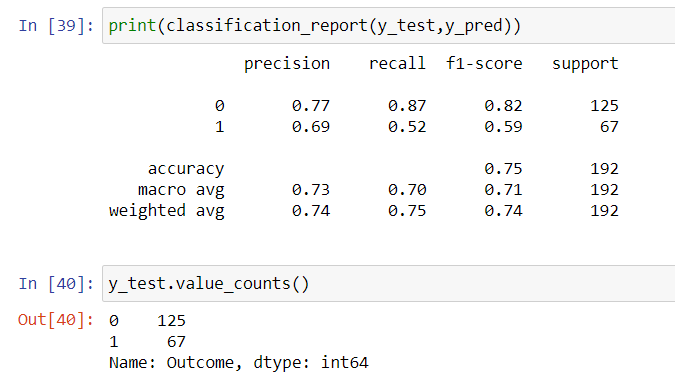
|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediction | | |
| Actual/Ground Reality |  | False | True |
| False | True Negativ (TN) | False Positive (FP) |
| True | False Negative (FN) | True Positive (TP) |

**Table 2: Confusion matrix from sklearn.metric**

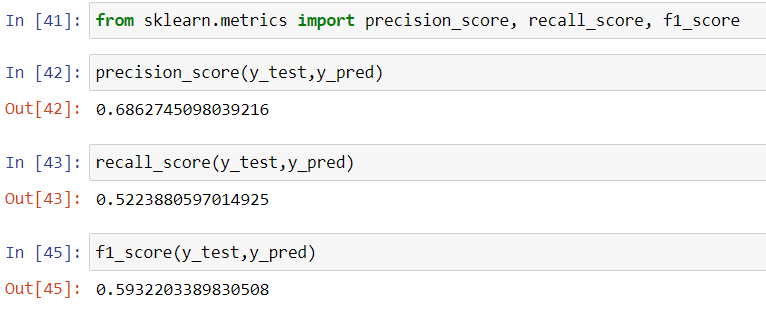
The test dataset used to obtain confusion matrix, has 125 non-diabetic datapoints and 65 diabetic datapoints and it is shown in Figure 1 along horizontal direction. Addition of numbers along vertical direction doesn’t match with the test dataset class numbers. Precision, recall and f1 score have also been studied to understand the discrepancy between the general form and sklearn.metrics library confusion matrix. The precision (77.3%) and recall (87.2%) values (refer Figure 1) calculated as per general form confusion matrix do not match with the precision (68.6%) and recall (52.2%) values obtained using sklearn.metrics library. Therefore, one must be careful in evaluating the model using confusion matrix and proper understanding of the classes is required before using it.



**Figure 1: Confusion matrix using sklearn.metrics library**



**Figure 2: Model evaluation summary using sklearn.metrics library**



**Figure 3: Precision, recall and f1\_score using sklearn.metrics library**