

A FALL 2019 ELEG 6163 PROJECT REPORT

TOPIC: BINARY CLASSIFIER WITH LOGISTIC REGRESSION

PRESENTED BY: FAITH NWOKOMA

LECTURER: XISHUANGE DONG

**Content**

Task description

Model description

Evaluation methods

Result analysis

**Task description**

Diabetes has become a rampaging health issue that have left many men impotent and caused disaster in many homes. Adverse diabetes weakens even the liver and shortens people life span. The good thing about this global epidemic is that early detection of this can lead to correction. There are several features that can be used to determine in a hospital if someone is diabetic or not. These features will form the features for this model.

The purpose of this project is to evaluate these features to predict if a person has diabetes or not. The model is designed to be a binary classification. Hence, a person can either have diabetes or not, and why logistic regression model is good fit for this prediction.

Logistic regression is a good statistical prediction model is used to determine the probability of occurrence of an event as either pass or fail, win/lose, alive/dead, healthy/sick and so on. Logistic regression is widely used in many medical applications and decision making.

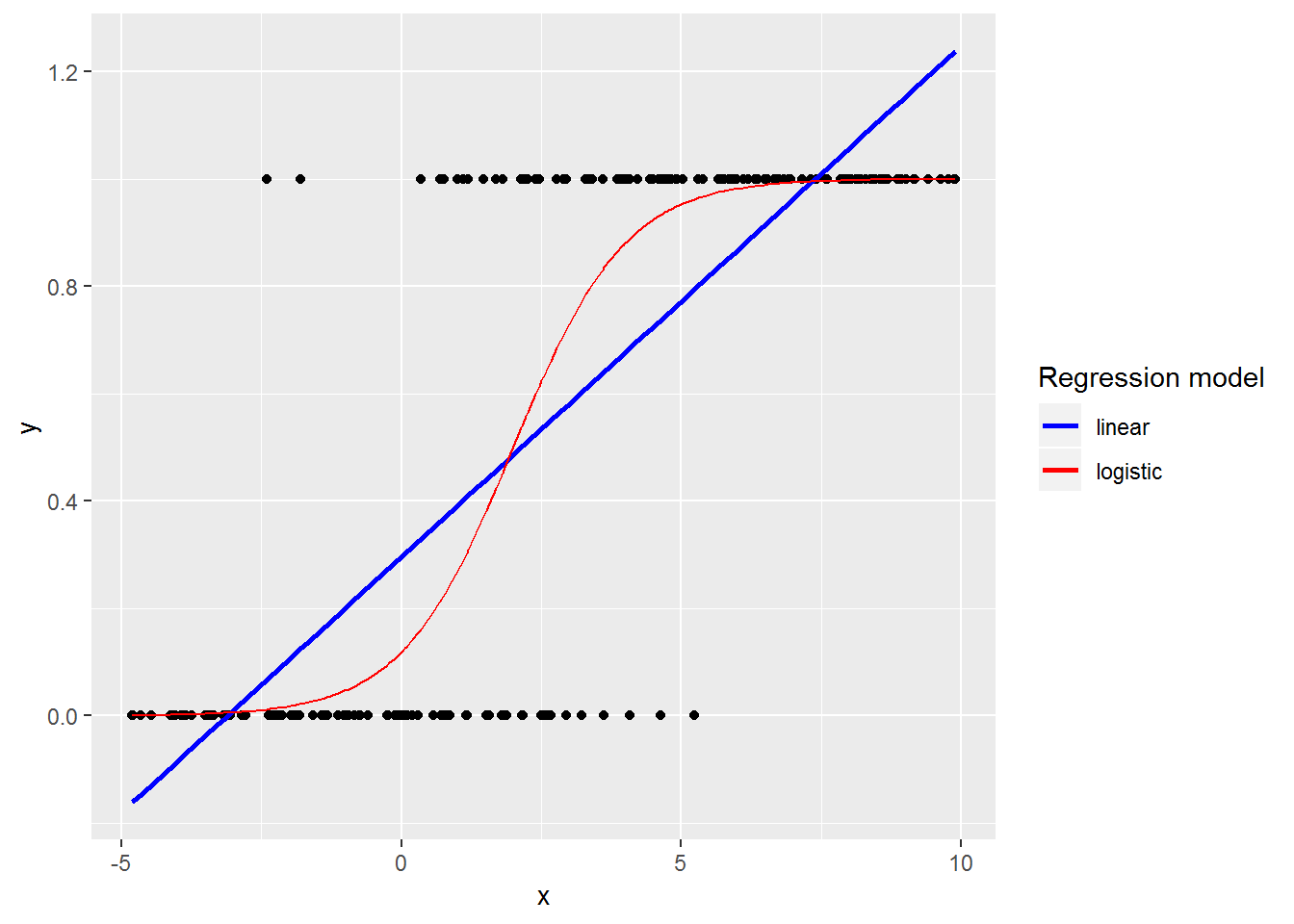


Fig. 1. Logistic regression with linear regression. Graph from bookdown.org – Logistic regression, chapter 6.

**Data Information**

The data is pima-indians-diabetes obtained from Github, supposedly the medical record of some Indian.

**Attributes:**

# 1. Number of times pregnant = pregnant

# 2. Plasma glucose concentration, a 2 hours in an oral glucose tolerance test = plasma\_glucose

# 3. Diastolic blood pressure (mm Hg) = DBP

# 4. Triceps skin fold thickness (mm) = skin\_thickness

# 5. 2-Hour serum insulin (mu U/ml) = insulin

# 6. Body mass index (weight in kg/(height in m)^2) = BMI

# 7. Diabetes pedigree function = diabetes\_function

# 8. Age (years) = age

Class (Label)

# 9. Class variable (0 or 1) = Target

Source

<https://gist.github.com/ktisha/c21e73a1bd1700294ef790c56c8aec1f>

**Model Description**

This will be simplified by stating the steps taken in building this model. It is not complex but rather understanding the purpose of the model will make it simple in deciding steps to follow.

**Step 1**. Downloading the data and converting the data into csv file in an excel format to understand the nature of the data and the features. This is where the feature names are decided and shortened for easy processing.

**Step 2.** The editor is prepared, libraries are imported, and features are separated from label

**Step 3.** Split the data into training and testing sets.

**Step 4.** The training and testing set are cleaned up, replacing missing values (zeros) with mean values to get D1 and D2 respectively.

**Step 5.** Feature selection is conducted on D1, with 5 feature selection, to obtain D3.

**Step 6.** I fit D2 also to 5 feature selection for D3 testing, hence D4.

**Step 7**. Prediction is performed on D4 with D3 model

**Step 8**. Training is also performed on D1. The dataset here has 8 features

**Step 9**. Prediction is performed on D2 with model trained with D1. The dataset here has 8 features

**Step 10**. To take my comparison further, Training is also performed on x\_train. This the raw data with missing values as 0s

**Step 11.** Prediction is performed on x\_test with model trained with x\_train. This the raw data with missing values as 0s.

**Evaluation Method**

The adopted evaluation methods are precision, recall and F-score.

**Precision** - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this performance evaluation metric tries to answer is of all persons tested and that are labeled as diabetic, how many are actually diabetic?

Precision = TP/TP+FP

**Recall**(Sensitivity) - Recall gives the ratio of correctly predicted positive observations to the all observations in actual class - Target. The question recall handles is: Of all the persons that really have diabetes, how many did I label?

Recall = TP/TP+FN

**F-score** – F-Score is the weighted average of Precision and Recall. As such, this takes both false positives and false negatives into account. F-score is usually more useful than accuracy, especially if you have a distribution of uneven class.

F-Score = 2\*(Recall \* Precision) / (Recall + Precision)

**Result Analysis**

Table 1. Prediction evaluation on D4 using D3 model

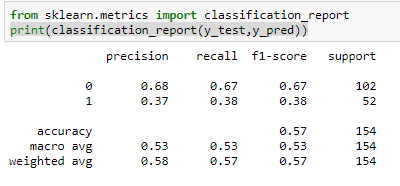


Table 2. Prediction evaluation on D2 using D1 model

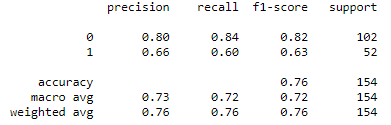
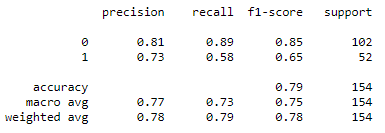


Table 3. Prediction evaluation on x\_test using x\_train model

­­­­­

From the tables above, you could see that the performance of the model is better with more features as can be seen between table two that has 8 features compared to table one with 5 selected features. Table three has the best scores. I think this is coming from the fact that zero values that were replaced with mean value probably may have degraded the result being that this is a sensitive classification that has to take a binary decision. Being that this is a medical report, data preprocessing is to be performed with expert judgement.

**Reference**

https://machinelearningmastery.com/feature-selection-machine-learning-python/

https://github.com/microsoft/LightGBM/blob/master/examples/python-guide/advanced\_example.py

https://github.com/bhattbhavesh91/GA\_Sessions/blob/master/ga\_dsmp\_5jan2019/16\_feature\_selection.ipynb

https://stackoverflow.com/questions/32011359/convert-categorical-data-in-pandas-dataframe/32011969

<https://stackoverflow.com/questions/11285613/selecting-multiple-columns-in-a-pandas-dataframe>

<https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>