Logistic Regression

Adwesh Behera - 20025006

11 September 2018

Problem Statement

The Pima Indians Diabetes Binary Classification dataset contains all of the data of female patients of the same age belonging to Pima Indian heritage. The data includes medical data, such as glucose and insulin levels, as well as lifestyle factors of the patients. The columns in the dataset are as follows: Number of times pregnant Plasma glucose concentration of 2 hours in an oral glucose tolerance test Diastolic blood pressure (mm Hg) Triceps skin fold thickness (mm) 2-hour serum insulin (mu U/ml) Body mass index (weight in kg/(height in m)^2) Diabetes pedigree function Age (years) Class variable (0 or 1) The last column is the target variable or class variable that takes the value 0 or 1, where 1 is positive or affected by diabetes and 0 means that the patient is not affected.

You have to build models that could predict whether a patient has diabetes or tests positive or not using logistic regression

Data

```
rm(list = ls())
data = read.csv("Pima Indians Diabetes Binary Classification dataset.csv")
str(data)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Number.of.times.pregnant                                                : int  6 1 8 1 0 5 3 10 2
##  $ Plasma.glucose.concentration.a.2.hours.in.an.oral.glucose.tolerance.test: int  148 85 183 89 137
##  $ Diastolic.blood.pressure..mm.Hg.                                        : int  72 66 64 66 40 74
##  $ Triceps.skin.fold.thickness..mm.                                        : int  35 29 0 23 35 0 32
##  $ X2.Hour.serum.insulin..mu.U.ml.                                         : int  0 0 0 94 168 0 88
##  $ Body.mass.index..weight.in.kg..height.in.m..2.                          : num  33.6 26.6 23.3 28.
##  $ Diabetes.pedigree.function                                              : num  0.627 0.351 0.672
##  $ Age..years.                                                             : int  50 31 32 21 33 30
##  $ Class.variable..0.or.1.                                                 : int  1 0 1 0 1 0 1 0 1
```

Division into Training and Test Datasets

We make take 100 rows as test cases and use the rest 668 as training data.

```
sample_i = sample(768, 100)
data_train = data[-sample_i,]
data_test = data[sample_i,]
```

Model

```
model = glm(Class.variable..0.or.1. ~ ., data = data_train, family = binomial)
model
```

```
##
## Call:  glm(formula = Class.variable..0.or.1. ~ ., family = binomial, 
##     data = data_train)
##
## Coefficients:
##                                                              (Intercept)
##                                                                -8.609944
##                                                 Number.of.times.pregnant
##                                                                 0.139365
## Plasma.glucose.concentration.a.2.hours.in.an.oral.glucose.tolerance.test
##                                                                 0.036185
##                                         Diastolic.blood.pressure..mm.Hg.
##                                                                -0.013975
##                                         Triceps.skin.fold.thickness..mm.
##                                                                -0.001150
##                                          X2.Hour.serum.insulin..mu.U.ml.
##                                                                -0.001104
##                           Body.mass.index..weight.in.kg..height.in.m..2.
##                                                                 0.089627
##                                               Diabetes.pedigree.function
##                                                                 1.154178
##                                                              Age..years.
##                                                                 0.013699
##
## Degrees of Freedom: 667 Total (i.e. Null);  659 Residual
## Null Deviance:       860.2
## Residual Deviance: 613.6     AIC: 631.6
```

Prediction

```
predict = ifelse(predict(model, data_test,type = "response") > 0.5, 1, 0)
```

Accuracy

```
table(predict, data_test$Class.variable..0.or.1.)
```

```
##
## predict  0  1
##       0 54 18
##       1  8 20
```

```
error = mean(predict != data_test$Class.variable..0.or.1.)
accuracy = 1 - error

print(accuracy)
```

```
## [1] 0.74
```