**What is Logistic Regression?**

Logistic regression is used for binary classification  where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 it belongs to Class 0. It’s referred to as regression because it is the extension of linear regression but is mainly used for classification problems.

**Key Points:**

* Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
* It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* In Logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).

**Logistic Function – Sigmoid Function**

* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the “S” form.
* The S-form curve is called the Sigmoid function or the logistic function.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0

## Assumptions of Logistic Regression

We will explore the assumptions of logistic regression as understanding these assumptions is important to ensure that we are using appropriate application of the model. The assumption include:

1. Independent observations: Each observation is independent of the other. meaning there is no correlation between any input variables.
2. Binary dependent variables: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.
3. Linearity relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.
4. No outliers: There should be no outliers in the dataset.
5. Large sample size: The sample size is sufficiently large

## ****Terminologies involved in Logistic Regression****

Here are some common terms involved in logistic regression:

* **Independent variables:** The input characteristics or predictor factors applied to the dependent variable’s predictions.
* **Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
* **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.
* **Odds:**It is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.
* **Log-odds:**The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.
* **Coefficient:**The logistic regression model’s estimated parameters, show how the independent and dependent variables relate to one another.
* **Intercept:**A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.
* **[Maximum likelihood estimation](https://www.geeksforgeeks.org/probability-density-estimation-maximum-likelihood-estimation/):** The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

PRACTICAL 10

Perform the logistic regression on the given data warehouse data.

To perform this use quality.csv file from following link:

#provide path of file where it is saved on your machine

quality <- read.csv('C:/quality.csv')

> #analysing the quality dataset

> str(quality)

'data.frame': 131 obs. of 14 variables:

$ MemberID : int 1 2 3 4 5 6 7 8 9 10 ...

$ InpatientDays : int 0 1 0 0 8 2 16 2 2 4 ...

$ ERVisits : int 0 1 0 1 2 0 1 0 1 2 ...

$ OfficeVisits : int 18 6 5 19 19 9 8 8 4 0 ...

$ Narcotics : int 1 1 3 0 3 2 1 0 3 2 ...

$ DaysSinceLastERVisit: num 731 411 731 158 449 ...

$ Pain : int 10 0 10 34 10 6 4 5 5 2 ...

$ TotalVisits : int 18 8 5 20 29 11 25 10 7 6 ...

$ ProviderCount : int 21 27 16 14 24 40 19 11 28 21 ...

$ MedicalClaims : int 93 19 27 59 51 53 40 28 20 17 ...

$ ClaimLines : int 222 115 148 242 204 156 261 87 98 66 ...

$ StartedOnCombination: logi FALSE FALSE FALSE FALSE FALSE FALSE ...

$ AcuteDrugGapSmall : int 0 1 5 0 0 4 0 0 0 0 ...

$ PoorCare : int 0 0 0 0 0 1 0 0 1 0 ...

> table(quality$PoorCare)

0 1

98 33

> 98/131

[1] 0.7480916

> install.packages("caTools")

Installing package into ‘C:/Users/Gauri/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

--- Please select a CRAN mirror for use in this session ---

also installing the dependency ‘bitops’

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/bitops\_1.0-6.zip'

Content type 'application/zip' length 38894 bytes (37 KB)

downloaded 37 KB

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/caTools\_1.17.1.1.zip'

Content type 'application/zip' length 329665 bytes (321 KB)

downloaded 321 KB

package ‘bitops’ successfully unpacked and MD5 sums checked

package ‘caTools’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Gauri\AppData\Local\Temp\RtmpmUN9oK\downloaded\_packages

> library(caTools)

Warning message:

package ‘caTools’ was built under R version 3.5.2

> set.seed(88)

> split = sample.split(quality$PoorCare, SplitRatio = 0.75)

>

> split

[1] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE

[28] TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE

[55] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[82] TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

[109] TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

> qualityTrain = subset(quality, split == TRUE)

> qualityTest = subset(quality, split == FALSE)

> nrow(qualityTrain)

[1] 99

> nrow(qualityTest)

[1] 32

> QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics,data=qualityTrain, family=binomial)

> summary(QualityLog)

Call:

glm(formula = PoorCare ~ OfficeVisits + Narcotics, family = binomial,

data = qualityTrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.06303 -0.63155 -0.50503 -0.09689 2.16686

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.64613 0.52357 -5.054 4.33e-07 \*\*\*

OfficeVisits 0.08212 0.03055 2.688 0.00718 \*\*

Narcotics 0.07630 0.03205 2.381 0.01728 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 111.888 on 98 degrees of freedom

Residual deviance: 89.127 on 96 degrees of freedom

AIC: 95.127

Number of Fisher Scoring iterations: 4

> predictTrain = predict(QualityLog, type="response")

> summary(predictTrain)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.06623 0.11912 0.15967 0.25253 0.26765 0.98456

> tapply(predictTrain, qualityTrain$PoorCare, mean)

0 1

0.1894512 0.4392246

> table(qualityTrain$PoorCare, predictTrain > 0.5)

FALSE TRUE

0 70 4

1 15 10

> 10/25

[1] 0.4

> 70/74

[1] 0.9459459

> table(qualityTrain$PoorCare, predictTrain > 0.7)

FALSE TRUE

0 73 1

1 17 8

> 8/25

[1] 0.32

> 73/74

[1] 0.9864865

> table(qualityTrain$PoorCare, predictTrain > 0.2)

FALSE TRUE

0 54 20

1 9 16

> 16/25

[1] 0.64

> 54/74

[1] 0.7297297

> install.packages("ROCR")

Installing package into ‘C:/Users/Gauri/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

also installing the dependencies ‘gtools’, ‘gdata’, ‘gplots’

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gtools\_3.8.1.zip'

Content type 'application/zip' length 325812 bytes (318 KB)

downloaded 318 KB

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gdata\_2.18.0.zip'

Content type 'application/zip' length 1260728 bytes (1.2 MB)

downloaded 1.2 MB

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/gplots\_3.0.1.1.zip' Content type 'application/zip' length 656764 bytes (641 KB)

downloaded 641 KB

trying URL 'http://mirror.its.dal.ca/cran/bin/windows/contrib/3.5/ROCR\_1.0-7.zip'

Content type 'application/zip' length 201823 bytes (197 KB)

downloaded 197 KB

package ‘gtools’ successfully unpacked and MD5 sums checked

package ‘gdata’ successfully unpacked and MD5 sums checked

package ‘gplots’ successfully unpacked and MD5 sums checked

package ‘ROCR’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Gauri\AppData\Local\Temp\RtmpmUN9oK\downloaded\_packages

> library(ROCR)

Loading required package: gplots

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

lowess

Warning messages:

1: package ‘ROCR’ was built under R version 3.5.2

2: package ‘gplots’ was built under R version 3.5.2

> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)

> ROCRperf = performance(ROCRpred, "tpr", "fpr")

> plot(ROCRperf)

> plot(ROCRperf, colorize=TRUE)

> plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

