R Notebook

Derrick Martin

09/25/2020

Logistic Regression is a linear method of classification where data is separated using a sigmoid function and all of the values are represented in the [0,1] range. The algorithm calculated the log odds from the estimated parameters and creates a boundary between classes using the variables provided. Logistic regression is good for classifying data that are linearly separable without using a lot of computational power, however it is prone to underfitting and does not work on non-linear decision boundaries.

Dividing the data into train and test

```
library(readr)
df <- read_csv("heart_disease_data.csv", show_col_types = FALSE)

set.seed(1)

sample <- sample(c(TRUE,FALSE), nrow(df), replace = TRUE, prob = c(0.80,0.20))

train <- df[sample, ]
test  <- df[!sample, ]
```

Data Exploration

This code is showing different information about the data so that we can get a better idea of the values and counts of different items.

```
names(train)
```

```
##  [1] "HeartDiseaseorAttack" "HighBP"              "HighChol"
##  [4] "CholCheck"            "BMI"                 "Smoker"
##  [7] "Stroke"               "Diabetes"            "PhysActivity"
## [10] "Fruits"               "Veggies"             "HvyAlcoholConsump"
## [13] "AnyHealthcare"        "NoDocbcCost"         "GenHlth"
## [16] "MentHlth"             "PhysHlth"            "DiffWalk"
## [19] "Sex"                  "Age"                 "Education"
## [22] "Income"
```

```
dim(train)
```

```
## [1] 202894     22
```

```
summary(train)
```

```
##  HeartDiseaseorAttack     HighBP         HighChol       CholCheck
##  Min.   :0.00000      Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:0.000   1st Qu.:0.000   1st Qu.:1.0000
##  Median :0.00000      Median :0.000   Median :0.000   Median :1.0000
##  Mean   :0.09398      Mean   :0.428   Mean   :0.424   Mean   :0.9628
##  3rd Qu.:0.00000      3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :1.000   Max.   :1.000   Max.   :1.0000
##       BMI            Smoker           Stroke           Diabetes
##  Min.   :12.00   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:24.00   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :27.00   Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :28.38   Mean   :0.4431   Mean   :0.04054   Mean   :0.2965
##  3rd Qu.:31.00   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :98.00   Max.   :1.0000   Max.   :1.00000   Max.   :2.0000
##   PhysActivity        Fruits          Veggies       HvyAlcoholConsump
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median :1.0000   Median :0.0000
##  Mean   :0.7562   Mean   :0.6344   Mean   :0.8116   Mean   :0.0564
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  AnyHealthcare     NoDocbcCost         GenHlth         MentHlth
##  Min.   :0.0000   Min.   :0.00000   Min.   :1.000   Min.   : 0.00
##  1st Qu.:1.0000   1st Qu.:0.00000   1st Qu.:2.000   1st Qu.: 0.00
##  Median :1.0000   Median :0.00000   Median :2.000   Median : 0.00
##  Mean   :0.9509   Mean   :0.08422   Mean   :2.511   Mean   : 3.19
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:3.000   3rd Qu.: 2.00
##  Max.   :1.0000   Max.   :1.00000   Max.   :5.000   Max.   :30.00
##     PhysHlth         DiffWalk           Sex              Age
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   : 1.00
##  1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 6.00
##  Median : 0.000   Median :0.0000   Median :0.0000   Median : 8.00
##  Mean   : 4.245   Mean   :0.1683   Mean   :0.4404   Mean   : 8.03
##  3rd Qu.: 3.000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:10.00
##  Max.   :30.000   Max.   :1.0000   Max.   :1.0000   Max.   :13.00
##    Education         Income
##  Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:5.000
##  Median :5.000   Median :7.000
##  Mean   :5.048   Mean   :6.051
##  3rd Qu.:6.000   3rd Qu.:8.000
##  Max.   :6.000   Max.   :8.000
```

```
str(train)
```

```
## tibble [202,894 × 22] (S3: tbl_df/tbl/data.frame)
##  $ HeartDiseaseorAttack: num [1:202894] 0 0 0 0 0 1 0 0 0 0 ...
##  $ HighBP              : num [1:202894] 1 0 1 1 1 1 0 0 1 0 ...
##  $ HighChol            : num [1:202894] 1 0 1 1 1 1 0 0 1 0 ...
##  $ CholCheck           : num [1:202894] 1 0 1 1 1 1 1 1 1 1 ...
##  $ BMI                 : num [1:202894] 40 25 28 24 25 30 24 25 34 26 ...
##  $ Smoker              : num [1:202894] 1 1 0 0 1 1 0 1 1 1 ...
##  $ Stroke              : num [1:202894] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Diabetes            : num [1:202894] 0 0 0 0 0 2 0 2 0 0 ...
##  $ PhysActivity        : num [1:202894] 0 1 0 1 1 0 0 1 0 0 ...
##  $ Fruits              : num [1:202894] 0 0 1 1 0 1 0 1 1 0 ...
##  $ Veggies             : num [1:202894] 1 0 0 1 1 1 1 1 1 1 ...
##  $ HvyAlcoholConsump   : num [1:202894] 0 0 0 0 0 0 0 0 0 0 ...
##  $ AnyHealthcare       : num [1:202894] 1 0 1 1 1 1 1 1 1 1 ...
##  $ NoDocbcCost         : num [1:202894] 0 1 1 0 0 0 0 0 0 0 ...
##  $ GenHlth             : num [1:202894] 5 3 5 2 3 5 2 3 3 3 ...
##  $ MentHlth            : num [1:202894] 18 0 30 3 0 30 0 0 0 0 ...
##  $ PhysHlth            : num [1:202894] 15 0 30 0 0 30 0 0 30 15 ...
##  $ DiffWalk            : num [1:202894] 1 0 1 0 1 1 0 0 1 0 ...
##  $ Sex                 : num [1:202894] 0 0 0 0 0 0 1 1 0 0 ...
##  $ Age                 : num [1:202894] 9 7 9 11 11 9 8 13 10 7 ...
##  $ Education           : num [1:202894] 4 6 4 5 4 5 4 6 5 5 ...
##  $ Income              : num [1:202894] 3 1 8 4 4 1 3 8 1 7 ...
```

```
head(train)
```

HeartDiseaseorAttack <dbl>	Hig… <dbl>	HighC… <dbl>	CholCh… <dbl>	… <dbl>	Sm… <dbl>	Stroke <dbl>	Diabetes <dbl>	PhysActivity <dbl>
0	1	1	1	40	1	0	0	0
0	0	0	0	25	1	0	0	1
0	1	1	1	28	0	0	0	0
0	1	1	1	24	0	0	0	1
0	1	1	1	25	1	0	0	1
1	1	1	1	30	1	0	2	0

6 rows | 1-10 of 22 columns

```
tail(train)
```

HeartDiseaseorAttack <dbl>	Hig… <dbl>	HighC… <dbl>	CholCh… <dbl>	… <dbl>	Sm… <dbl>	Stroke <dbl>	Diabetes <dbl>	PhysActivity <dbl>
0	0	0	1	27	0	0	0	0
0	1	1	1	45	0	0	0	0

HeartDiseaseorAttack <dbl>	Hig… <dbl>	HighC… <dbl>	CholCh… <dbl>	… <dbl>	Sm… <dbl>	Stroke <dbl>	Diabetes <dbl>	PhysActivity <dbl>
0	1	1	1	18	0	0	2	0
0	0	0	1	28	0	0	0	1
0	1	0	1	23	0	0	0	0
1	1	1	1	25	0	0	2	1

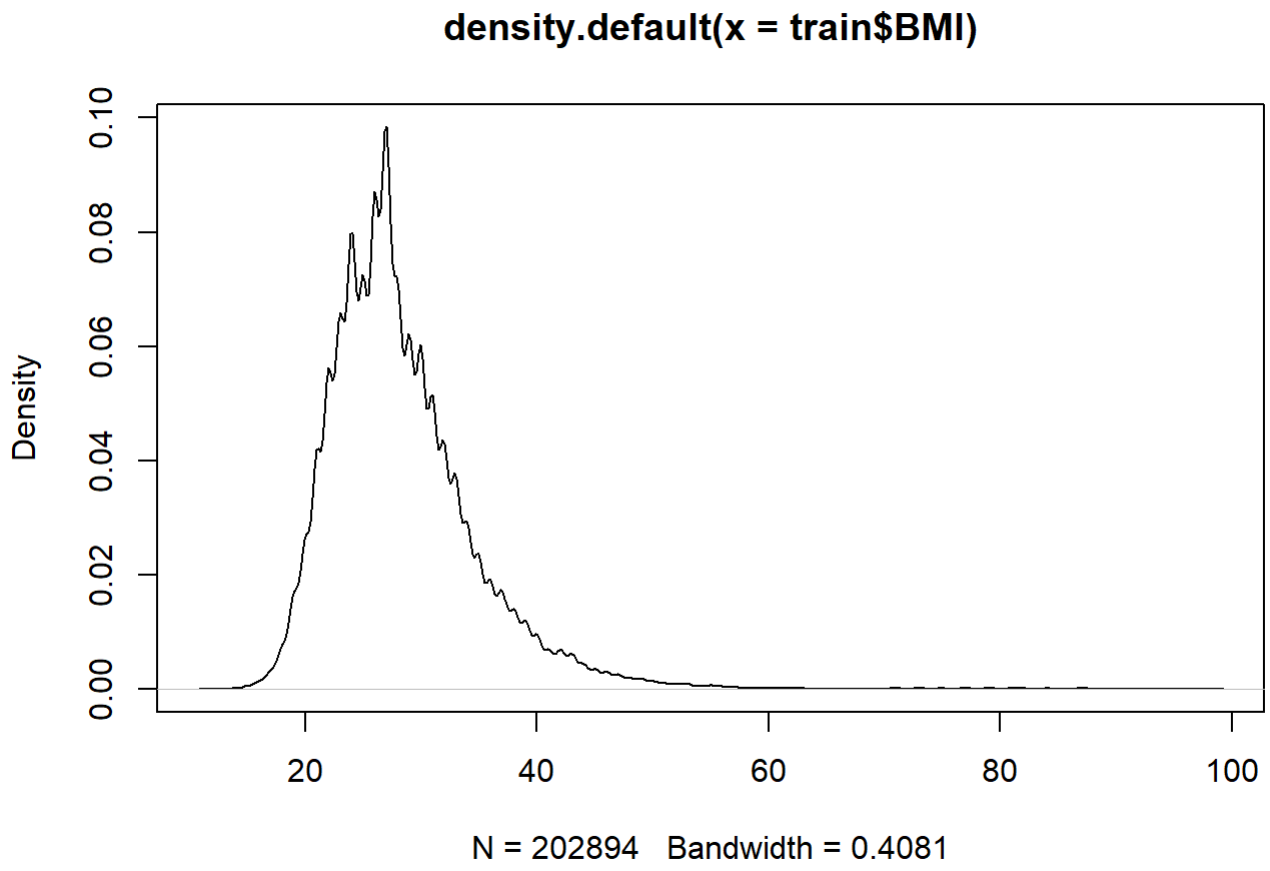
6 rows | 1-10 of 22 columns

Graphs

In this data set the Age is broken into age categories numbered 1-14. The age categories are as follows.

1. Age 18 to 24
2. Age 25 to 29
3. Age 30 to 34
4. Age 35 to 39
5. Age 40 to 44
6. Age 45 to 49
7. Age 50 to 54
8. Age 55 to 59
9. Age 60 to 64
10. Age 65 to 69
11. Age 70 to 74
12. Age 75 to 79
13. Age 80 or older

```
d <- density(train$BMI)
plot(d)
```




```
hist(train$Age)
```

Building Simple Linear Regression model

In the summary we can see that the Residual deviance is somewhat similar to the null deviance which is not good as we want the RD to be smaller as that would indicate a better fit for the model. The high AIC also indicates that there is not a preference for simpler models with less predictors.

```
model1 <- glm(HeartDiseaseorAttack~ BMI + Age, data = train, family = binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = HeartDiseaseorAttack ~ BMI + Age, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7029  -0.4971  -0.3556  -0.2157   3.2633
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.318315   0.050760 -124.47   <2e-16 ***
## BMI          0.037281   0.001083   34.43   <2e-16 ***
## Age          0.327488   0.003383   96.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 126469  on 202893  degrees of freedom
## Residual deviance: 114014  on 202891  degrees of freedom
## AIC: 114020
##
## Number of Fisher Scoring iterations: 6
```

Naive Bayes model

Here we can see that the NB predicts that based on all of the factors HeartDiseaseorAttack can be shown not to be present with around 90 percent accuracy. There is then a breakdown of all of the factors and how they individually can or can not predict HeartDiseaseorAttack or the lack thereof.

```
library(e1071)
nb1 <- naiveBayes(HeartDiseaseorAttack~., data = train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1
## 0.90601496 0.09398504
##
## Conditional probabilities:
##    HighBP
## Y        [,1]      [,2]
##   0 0.3945492 0.4887550
##   1 0.7500656 0.4329862
##
##    HighChol
## Y        [,1]      [,2]
##   0 0.3949626 0.4888440
##   1 0.7036027 0.4566803
##
##    CholCheck
## Y        [,1]      [,2]
##   0 0.9601632 0.1955762
##   1 0.9882532 0.1077471
##
##    BMI
## Y       [,1]     [,2]
##   0 28.26502 6.579640
##   1 29.44176 6.685127
##
##    Smoker
## Y        [,1]      [,2]
##   0 0.4246648 0.4942934
##   1 0.6206408 0.4852403
##
##    Stroke
## Y         [,1]      [,2]
##   0 0.02779274 0.1643790
##   1 0.16340658 0.3697459
##
##    Diabetes
## Y        [,1]      [,2]
##   0 0.2557106 0.6548379
##   1 0.6892863 0.9353449
##
##    PhysActivity
## Y        [,1]      [,2]
##   0 0.7683830 0.4218667
##   1 0.6382611 0.4805164
##
```

```
##    Fruits
## Y        [,1]      [,2]
##   0 0.6377152 0.4806618
##   1 0.6019718 0.4895042
##
##    Veggies
## Y        [,1]      [,2]
##   0 0.8163743 0.3871797
##   1 0.7651686 0.4239045
##
##    HvyAlcoholConsump
## Y         [,1]      [,2]
##   0 0.05854481 0.2347714
##   1 0.03576485 0.1857082
##
##    AnyHealthcare
## Y        [,1]      [,2]
##   0 0.9496206 0.2187272
##   1 0.9628192 0.1892095
##
##    NoDocbcCost
## Y         [,1]      [,2]
##   0 0.08139807 0.2734462
##   1 0.11138497 0.3146165
##
##    GenHlth
## Y       [,1]     [,2]
##   0 2.422614 1.026579
##   1 3.367979 1.087066
##
##    MentHlth
## Y       [,1]     [,2]
##   0 3.032194 7.183730
##   1 4.706330 9.232242
##
##    PhysHlth
## Y       [,1]      [,2]
##   0 3.739475  8.165534
##   1 9.115370 11.859821
##
##    DiffWalk
## Y        [,1]      [,2]
##   0 0.1428369 0.3499074
##   1 0.4133935 0.4924551
##
##    Sex
## Y        [,1]      [,2]
##   0 0.4265579 0.4945782
##   1 0.5739682 0.4945114
##
##    Age
## Y        [,1]     [,2]
```

```
##   0  7.811571 3.050760
##   1 10.135613 2.222179
##
##    Education
## Y       [,1]      [,2]
##   0 5.080131 0.9725978
##   1 4.743353 1.0641765
##
##    Income
## Y       [,1]     [,2]
##   0 6.144289 2.037039
##   1 5.146521 2.200375
```

Predictions

According to these results the NB model predicts with roughly an 80 percent accuracy while the LR model has 23 percent correlation on the predictions. The NB predicted much more accurately I think because it takes all of the variables into account while the LR model only attempts to predict based on two variables.

```
pred1 <- predict(model1, newdata = test)
pred2 <- predict(nb1, newdata = test, type = "class")
mean(pred2==test$HeartDiseaseorAttack)
```

```
## [1] 0.8202654
```

```
cor(pred1,test$HeartDiseaseorAttack)
```

```
## [1] 0.2303157
```

Strengths and weaknesses

Niave Bayes is better at evaluating smaller data sets and is pretty easy to implement and interpret but can be outperformed by other classifiers on larger data sets and the niave assumption that the predictors are independent may be wrong. Logistic Regression is good for classifying data that are linearly separable without using a lot of computational power, however it is prone to underfitting and does not work on non-linear decision boundaries.