Tick-Borne_illness_stats

ChiChi Ugochukwu

2022-10-31

Conducting Statistical Analysis on the Tick-Borne Illness Dataset

In an effort to garner more information from our data, we run a logistic regression test to see how much influence the individual symptoms have on the test result.

Loading in the data

The data set was exported from our PostgreSQL database as a csv onto our local drive before being loaded.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyr)
library(ggplot2)

animalData <- read.csv(file = "ml_clean_animalData.csv")
head(animalData)

##   index         age weight_lbs temperature heart_rate_bpm resp_rate_bpm mm
crt
## 1     6 1.000000000         15         102            183            44 2
0
## 2     8 2.000000000         72         100            138            21 2
2
## 3    10 0.005479452         63         102            160            50 2
0
## 4    11 0.005479452         59         102            157            18 2
2
## 5    19 6.000000000         42         104            110            40 0
0
```

```
## 6    20 0.166666667         90         100            161           131  2
2
##   mentation vomiting diarrhea inappetence lethargic lameness muscle_pain
## 1         4        1        1           2         2        0           2
## 2         4        1        3           2         2        0           2
## 3         4        3        3           0         2        0           2
## 4         2        1        1           1         2        0           2
## 5         2        3        2           0         2        0           0
## 6         0        3        3           2         2        0           2
##   joint_swelling reported_weight_loss skin_condition is_4dx_tested
## 1              2                    0              2      Negative
## 2              2                    1              2      Negative
## 3              2                    0              2      Negative
## 4              0                    0              2      Negative
## 5              0                    0              2      Positive
## 6              2                    0              2      Negative
```

```
colnames(animalData)
```

```
##  [1] "index"                "age"                  "weight_lbs"
##  [4] "temperature"          "heart_rate_bpm"       "resp_rate_bpm"
##  [7] "mm"                   "crt"                  "mentation"
## [10] "vomiting"             "diarrhea"             "inappetence"
## [13] "lethargic"            "lameness"             "muscle_pain"
## [16] "joint_swelling"       "reported_weight_loss" "skin_condition"
## [19] "is_4dx_tested"
```

Below, we perform a cursory check to make sure the data columns have been loaded correctly and to see what the data types are to make changes before the analysis.

```
str(animalData)
```

```
## 'data.frame':    16979 obs. of  19 variables:
##  $ index               : int  6 8 10 11 19 20 24 25 32 35 ...
##  $ age                 : num  1 2 0.00548 0.00548 6 ...
##  $ weight_lbs          : int  15 72 63 59 42 90 19 71 47 69 ...
##  $ temperature         : int  102 100 102 102 104 100 102 102 99 105 ...
##  $ heart_rate_bpm      : int  183 138 160 157 110 161 88 147 93 178 ...
##  $ resp_rate_bpm       : int  44 21 50 18 40 131 150 39 39 145 ...
##  $ mm                  : int  2 2 2 2 0 2 2 2 2 2 ...
##  $ crt                 : int  0 2 0 2 0 2 1 1 0 0 ...
##  $ mentation           : int  4 4 4 2 2 0 2 1 4 4 ...
##  $ vomiting            : int  1 1 3 1 3 3 3 1 3 1 ...
##  $ diarrhea            : int  1 3 3 1 2 3 3 3 1 3 ...
##  $ inappetence         : int  2 2 0 1 0 2 1 0 0 0 ...
##  $ lethargic           : int  2 2 2 2 2 2 2 0 2 0 ...
##  $ lameness            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ muscle_pain         : int  2 2 2 2 0 2 2 2 0 2 ...
##  $ joint_swelling      : int  2 2 2 0 0 2 2 2 2 0 ...
##  $ reported_weight_loss: int  0 1 0 0 0 0 0 0 0 1 ...
##  $ skin_condition      : int  2 2 2 2 2 2 0 2 2 2 ...
##  $ is_4dx_tested       : chr  "Negative" "Negative" "Negative" "Negative"
...
```

Looking at the data, we can see that many of the columns need to be encoded as factors instead of integers. To make this change, we execute the code below:

```
animalData <- subset(animalData, select = -c(index))
animalData$mm <- as.factor(animalData$mm)
animalData$crt <- as.factor(animalData$crt)
animalData$mentation <- as.factor(animalData$mentation)
animalData$vomiting <- as.factor(animalData$vomiting)
animalData$diarrhea <- as.factor(animalData$diarrhea)
animalData$inappetence <- as.factor(animalData$inappetence)
animalData$lethargic <- as.factor(animalData$lethargic)
animalData$muscle_pain <- as.factor(animalData$muscle_pain)
animalData$joint_swelling <- as.factor(animalData$joint_swelling)
animalData$skin_condition <- as.factor(animalData$skin_condition)
animalData$lameness <- as.factor(animalData$lameness)
animalData$reported_weight_loss <- as.factor(animalData$reported_weight_loss)
animalData$is_4dx_tested <- as.factor(animalData$is_4dx_tested)

str(animalData)
```

```
## 'data.frame':    16979 obs. of  18 variables:
##  $ age                 : num  1 2 0.00548 0.00548 6 ...
##  $ weight_lbs          : int  15 72 63 59 42 90 19 71 47 69 ...
##  $ temperature         : int  102 100 102 102 104 100 102 102 99 105 ...
##  $ heart_rate_bpm      : int  183 138 160 157 110 161 88 147 93 178 ...
##  $ resp_rate_bpm       : int  44 21 50 18 40 131 150 39 39 145 ...
##  $ mm                  : Factor w/ 3 levels "0","1","2": 3 3 3 3 1 3 3 3 3
3 ...
##  $ crt                 : Factor w/ 4 levels "0","1","2","3": 1 3 1 3 1 3 2
2 1 1 ...
##  $ mentation           : Factor w/ 5 levels "0","1","2","3",..: 5 5 5 3 3
1 3 2 5 5 ...
##  $ vomiting            : Factor w/ 4 levels "0","1","2","3": 2 2 4 2 4 4 4
2 4 2 ...
##  $ diarrhea            : Factor w/ 4 levels "0","1","2","3": 2 4 4 2 3 4 4
4 2 4 ...
##  $ inappetence         : Factor w/ 4 levels "0","1","2","3": 3 3 1 2 1 3 2
1 1 1 ...
##  $ lethargic           : Factor w/ 3 levels "0","1","2": 3 3 3 3 3 3 3 1 3
1 ...
##  $ lameness            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1
...
##  $ muscle_pain         : Factor w/ 3 levels "0","1","2": 3 3 3 3 1 3 3 3 1
3 ...
##  $ joint_swelling      : Factor w/ 3 levels "0","1","2": 3 3 3 1 1 3 3 3 3
1 ...
##  $ reported_weight_loss: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 2
...
```

```
##  $ skin_condition      : Factor w/ 5 levels "0","1","2","3",..: 3 3 3 3 3
3 1 3 3 3 ...
##  $ is_4dx_tested       : Factor w/ 2 levels "Negative","Positive": 1 1 1 1
2 1 1 1 1 2 ...
```

Just briefly - for understanding - this is what the factor levels now mean:

- mm [1:‘Light Pink’, 2:‘Pale’, 3:‘Pink’]
- crt [1:‘1-2 sec’, 2:‘<1 sec’, 3:‘>2 sec’, 4:‘UTO’]
- mentation [1:‘Anxious/Agitated’, 2:‘BAR’, 3:‘Dull/Depressed’, 4:‘Obtunded’, 5:‘QAR’]
- vomiting [1:‘Chronic’, 2:‘Mild’, 3:‘Moderate’, 4:‘None’]
- diarrhea [1:‘Chronic’, 2:‘Mild’, 3:‘Moderate’, 4:‘None’]
- inappetence [1:‘Mild’, 2:‘Moderate’, 3:‘None’, 4:‘Severe’]
- lethargic [1:‘Mild’, 2:‘Moderate’, 3:‘None’]
- muscle pain [1:‘Mild’, 2:‘Moderate’, 3:‘None’]
- lameness [1:‘None’, 2:‘Present’]
- reported weight loss [1:‘None’, 2:‘Present’]
- joint swelling [1:‘Mild’, 2:‘Moderate’, 3:‘None’]
- skin condition [1:‘Bruising, 2:’Irritation’, 3:‘Normal’, 4:‘Petechia’, 5:‘Petechiae’]

Now that the data types are adjusted, we can proceed with the analysis.

```
logistic <- glm(is_4dx_tested ~ ., data=animalData, family="binomial")

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(logistic)

##
## Call:
## glm(formula = is_4dx_tested ~ ., family = "binomial", data = animalData)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -9.742e-05  -2.100e-08  -2.100e-08  -2.100e-08   1.520e-04
##
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -4.289e+03  1.935e+05  -0.022    0.982
## age                      1.547e-02  2.706e+02   0.000    1.000
## weight_lbs               3.398e-03  2.692e+01   0.000    1.000
## temperature              4.182e+01  1.888e+03   0.022    0.982
## heart_rate_bpm           2.852e-03  3.063e+01   0.000    1.000
## resp_rate_bpm            1.130e-03  1.977e+01   0.000    1.000
## mm1                     -5.561e-01  3.706e+03   0.000    1.000
## mm2                      5.054e-02  2.614e+03   0.000    1.000
## crt1                    -8.145e-02  2.359e+03   0.000    1.000
## crt2                  -2.130e-02  2.682e+03   0.000    1.000
## crt3                          NA         NA      NA       NA
## mentation1             8.593e-01  1.108e+04   0.000    1.000
## mentation2             1.629e+00  1.102e+04   0.000    1.000
## mentation3             1.097e+00  2.011e+04   0.000    1.000
## mentation4             7.700e-01  1.112e+04   0.000    1.000
## vomiting1              4.849e-01  6.088e+03   0.000    1.000
## vomiting2              7.132e-02  6.805e+03   0.000    1.000
## vomiting3              5.961e-01  6.077e+03   0.000    1.000
## diarrhea1              1.316e-01  6.109e+03   0.000    1.000
## diarrhea2             -3.357e-02  6.542e+03   0.000    1.000
## diarrhea3              3.459e-01  6.097e+03   0.000    1.000
## inappetence1          -5.742e-01  2.844e+03   0.000    1.000
## inappetence2          -1.129e+00  2.134e+03  -0.001    1.000
## inappetence3           6.072e-01  4.524e+03   0.000    1.000
## lethargic1            -2.197e-01  2.543e+03   0.000    1.000
## lethargic2            -9.243e-01  2.178e+03   0.000    1.000
## lameness1              2.244e+00  2.672e+03   0.001    0.999
## muscle_pain1           1.681e+00  3.321e+03   0.001    1.000
## muscle_pain2          -1.193e+00  2.149e+03  -0.001    1.000
## joint_swelling1        1.789e+00  3.461e+03   0.001    1.000
## joint_swelling2       -1.048e+00  2.069e+03  -0.001    1.000
## reported_weight_loss1  1.226e+00  2.125e+03   0.001    1.000
## skin_condition1       -1.170e+00  6.900e+03   0.000    1.000
## skin_condition2       -5.797e-01  2.263e+03   0.000    1.000
## skin_condition3        3.382e+00  1.472e+04   0.000    1.000
## skin_condition4       -6.783e-01  1.600e+04   0.000    1.000
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1.2808e+04  on 16978  degrees of freedom
## Residual deviance: 9.2589e-07  on 16944  degrees of freedom
## AIC: 70
## 
## Number of Fisher Scoring iterations: 25
```

In running the general logistic model, we see that none of the variables have produced a p-value below the significance level. In fact, for all of the variables, the p-value is equal or very close to 1. The program also warns that some of the data is too good of a predictor for our target. This could be the major influence that skewed our results.

Let's look at how the model does when we compare it to a single variable:

```
logistic2 <- glm(is_4dx_tested ~ vomiting, data=animalData,
family="binomial")

summary(logistic2)

## 
## Call:
```

```
## glm(formula = is_4dx_tested ~ vomiting, family = "binomial",
##     data = animalData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5682  -0.5218  -0.5106  -0.5106   2.0618
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.74175    0.09775 -17.819   <2e-16 ***
## vomiting1   -0.18366    0.10430  -1.761   0.0782 .
## vomiting2   -0.25670    0.11944  -2.149   0.0316 *
## vomiting3   -0.22966    0.10405  -2.207   0.0273 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12808  on 16978  degrees of freedom
## Residual deviance: 12803  on 16975  degrees of freedom
## AIC: 12811
##
## Number of Fisher Scoring iterations: 4
```

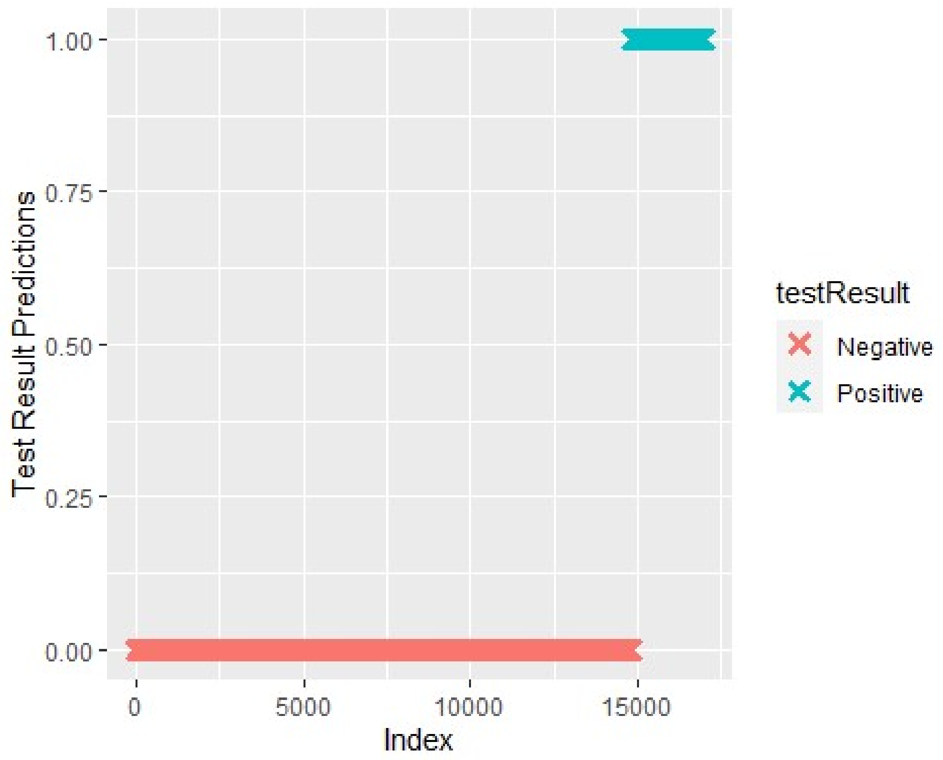
Here we see that the the variable ‘vomiting’ may be influential in predicting our target, where factors 2, 3, and 4 gives a p-value less than 0.05.

This indicates that were we to test each individual variable, we’d gain more insight from the data. However, due to time constraints, we are unable to do this in this analysis.

Plotting the results

For practice, we will also display the results of the logistic regression graphically.

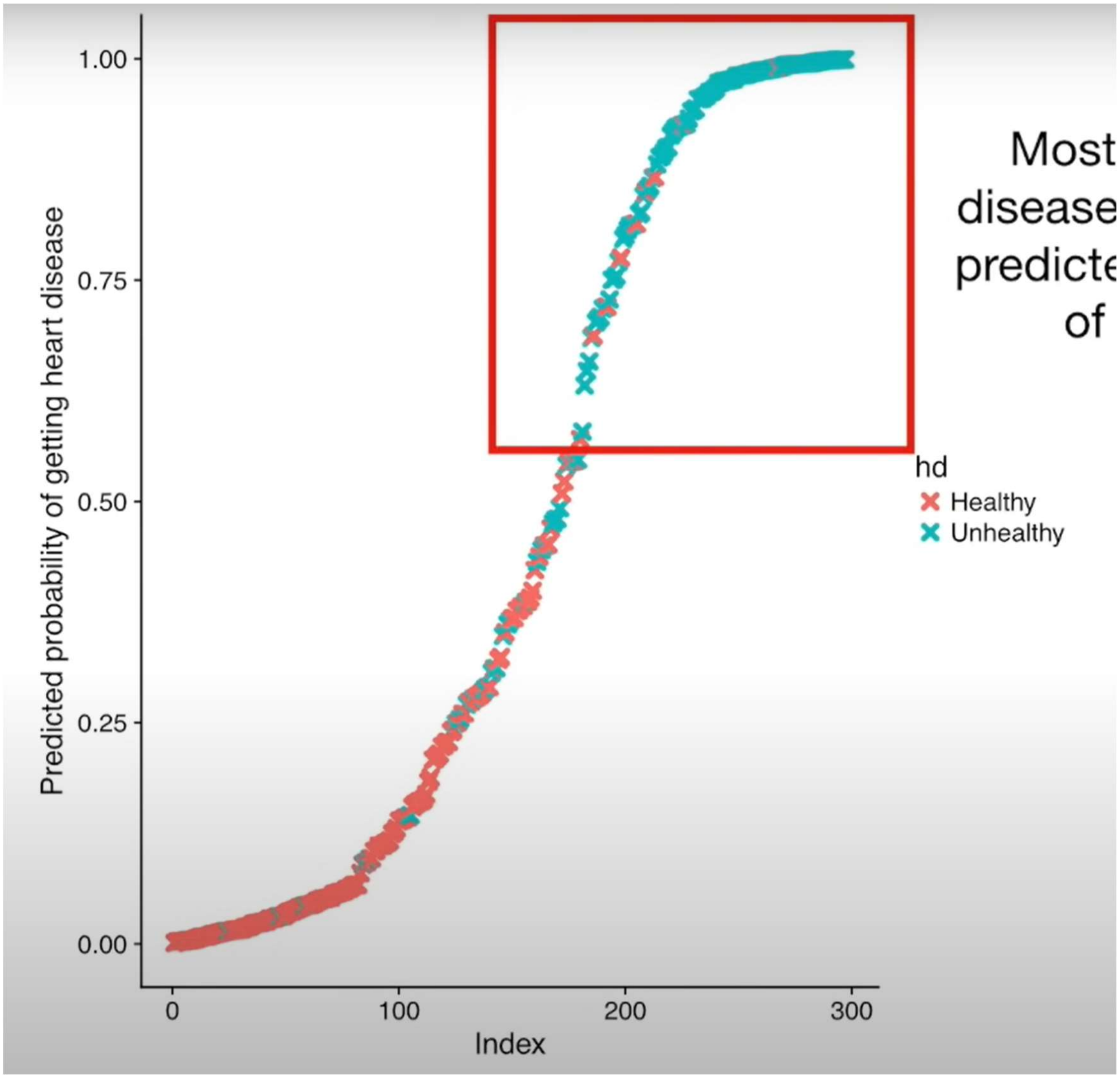
```
predicted.data <- data.frame(
  probability.of.testResult=logistic$fitted.values,
  testResult=animalData$is_4dx_tested)

predicted.data <- predicted.data[
  order(predicted.data$probability.of.testResult,
        decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)

ggplot(data=predicted.data, aes(x=rank, y=probability.of.testResult)) +
  geom_point(aes(color=testResult), alpha=1, shape=4, stroke=2) +
  xlab("Index") +
  ylab("Test Result Predictions")
```



```
ggsave("Test_result_prediction.png")
```

```
## Saving 5 x 4 in image
```

This graph further indicates that the data was too good of a predictor of the outcomes - typically we would see the signature S-shaped curve of logistic regressions like in the example phote shown below.



Example of logistic curve

Conclusion

In summary, this R analysis proved insightful. However, more work needs to be done to see if any useful results can be garnered.