CSCI 385 - Second Deliverable

Garrett Welton

11/15/2020

Predictions

Based upon my initial discovery and exploration I would like to predict the presence of heart disease in hospital patients from their medical data.

"Success" and "Failure"

For the predictions I am trying to make "Success" would be creating a model that accurately predicts the target variable or the presence of heart disease in a patient with more than 71%. "Failure" would be if my model was unable to accurately predict this.

New Data

My additional dataset is the "National Health Interview Survey (NHIS) - National Cardiovascular Disease Surveillance Data" published by the Centers for Disease Control and Prevention. The NHIS monitors the health of the nation through personal household interviews. This dataset is provided by the National Cardiovascular Disease Surveillance System, which is designed to integrate multiple indicators from many data sources to provide a comprehensive picture of the public health burden of CVDs and associated risk factors in the United States. This data is from 2001 and forward. This dataset should be useful for determining how different regions of the US affect the rate of cardiovascular disease within its different populations, as well as how different risk factors play into these rates. This new data will help me help me be able to make new and better predictions.

```
#API call
df <- read.socrata(
  "https://chronicdata.cdc.gov/resource/fwns-azgu.csv",
  app_token = "OJxxBUd1kEfoUDA5ACIVnD88u"
)
```

Tidying Data:

```
NHIS <- filter(df, !is.na(data_value), data_value_type == "Age-Standardized" | break_out_category == "Ag
  select(year, locationdesc, category, topic, data_value, break_out_category, break_out) %>%
  rename(location = locationdesc, percent_value = data_value)
```

I first filtered the data to get rid of any NA values and any non-age-standardized data, that was not a part of the age break_out_category. Then I used select to get rid of the unneeded columns. Finally I renamed some columns to make it clear what they are.

```r
cardioDiseases <- filter(NHIS, category == "Cardiovascular Diseases") %>%
  select(-category)

riskFactors <- filter(NHIS, category != "Cardiovascular Diseases") %>%
  select(-category)

head(cardioDiseases)
```

```
##   year      location                         topic percent_value
## 1 2000 United States Major Cardiovascular Disease           7.1
## 2 2000 United States Major Cardiovascular Disease           8.6
## 3 2000 United States Major Cardiovascular Disease           5.8
## 4 2000 United States Major Cardiovascular Disease           1.2
## 5 2000 United States Major Cardiovascular Disease           8.0
## 6 2000 United States Major Cardiovascular Disease          24.5
##   break_out_category break_out
## 1            Overall   Overall
## 2             Gender      Male
## 3             Gender    Female
## 4                Age     25-44
## 5                Age     45-64
## 6                Age       65+
```

```r
head(riskFactors)
```

```
##   year      location               topic percent_value break_out_category
## 1 2000 United States Physical Inactivity          37.8            Overall
## 2 2000 United States Physical Inactivity          35.0             Gender
## 3 2000 United States Physical Inactivity          40.3             Gender
## 4 2000 United States Physical Inactivity          29.4                Age
## 5 2000 United States Physical Inactivity          32.9                Age
## 6 2000 United States Physical Inactivity          40.4                Age
##   break_out
## 1   Overall
## 2      Male
## 3    Female
## 4     18-24
## 5     25-44
## 6     45-64
```

```r
cardioDiseases <- cardioDiseases %>%
  pivot_wider(names_from = c(break_out_category, break_out),
              values_from = percent_value)

colSums(is.na(cardioDiseases))
```

```
##                    year                location                   topic
##                       0                       0                       0
##         Overall_Overall             Gender_Male           Gender_Female
##                       0                       0                       0
##               Age_25-44               Age_45-64                 Age_65+
##                     144                       1                       0
```

```
##                  Age_35+                  Age_75+ Race_Non-Hispanic White
##                        0                        0                        0
## Race_Non-Hispanic Black Race_Non-Hispanic Asian           Race_Hispanic
##                       50                      236                       70
##              Race_Other                Age_18-24
##                      235                      262
```

```
cardioDiseases <- select(cardioDiseases, -`Age_25-44`, -`Age_18-24`, -Race_Other, -`Race_Non-Hispanic As
  drop_na()

unique(cardioDiseases$topic)
```

```
## [1] "Major Cardiovascular Disease"
## [2] "Acute Myocardial Infarction (Heart Attack)"
## [3] "Coronary Heart Disease"
## [4] "Stroke"
```

```
cardioDiseases.majorCardiovascularDisease <- filter(cardioDiseases, topic == "Major Cardiovascular Disea
  select(-topic)
cardioDiseases.heartAttack <- filter(cardioDiseases, topic == "Acute Myocardial Infarction (Heart Attack
  select(-topic)
cardioDiseases.coronaryHeartDisease <- filter(cardioDiseases, topic == "Coronary Heart Disease") %>%
  select(-topic)
cardioDiseases.stroke <- filter(cardioDiseases, topic == "Stroke") %>%
  select(-topic)

riskFactors <- riskFactors %>%
  pivot_wider(names_from = c(break_out_category, break_out),
              values_from = percent_value,
              values_fn = mean)

colSums(is.na(riskFactors))
```

```
##                    year                location                   topic
##                       0                       0                       0
##         Overall_Overall             Gender_Male           Gender_Female
##                       0                       0                       0
##               Age_18-24               Age_25-44               Age_45-64
##                       0                       0                       0
##                  Age_65+                  Age_35+                  Age_75+
##                       0                       0                       1
## Race_Non-Hispanic White Race_Non-Hispanic Black Race_Non-Hispanic Asian
##                       0                       0                       8
##           Race_Hispanic              Race_Other
##                       0                      34
```

```
riskFactors <- select(riskFactors, -Race_Other, -`Race_Non-Hispanic Asian`) %>%
  drop_na()

riskFactors.physicalInactivity <- filter(riskFactors, topic == "Physical Inactivity") %>%
  select(-topic)
riskFactors.smoking <- filter(riskFactors, topic == "Smoking") %>%
```

```
  select(-topic)
riskFactors.hypertension <- filter(riskFactors, topic == "Hypertension") %>%
  select(-topic)
```

I tried multiple methods to tidy my new data, however, the structure of the data seems to be a standard of the CDC and proved to be a challenge. Most of the publicly available datasets I found on cardiovascular disease were published by the CDC and shared this same format. The data contains many nested variable within its column variables, making it not tidy. The challenge is that most of the variable within these column are dependent on variables that are stored with other columns. Due to this I couldn't find a nice way to tidy the data. I tried splitting the data into multiple tables, but believe this can make working with the data more of a challenge.

Tables

```
head(NHIS)
```

```
##   year      location                category                          topic
## 1 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
## 2 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
## 3 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
## 4 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
## 5 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
## 6 2000 United States Cardiovascular Diseases Major Cardiovascular Disease
##   percent_value break_out_category break_out
## 1           7.1            Overall   Overall
## 2           8.6             Gender      Male
## 3           5.8             Gender    Female
## 4           1.2                Age     25-44
## 5           8.0                Age     45-64
## 6          24.5                Age       65+
```

- `year` - `integer` - year the data was collected.
- `location` - `character` - categorical variable containing the region of the US that the data was collected. (United States; Northeast; Midwest; South; West)
- `category` - `character` - variable describing the the category of the topic variable. (Cardiovascular Diseases, Risk Factor)
- `topic` - `character` - categorical variable describing the topic that the percent_value corresponds to. (Major Cardiovascular Disease; Acute Myocardial Infarction (Heart Attack); Coronary Heart Disease; Stroke; Physical Inactivity; Smoking; Hypertension)
- `percent_value` - `double` - the percent of the beak_out variable where topic variable is true. All values besides those that correspond to the "Age" break_out_category have been age-Standardized between different regions.
- `break_out_category` - `character` - variable describing the category of the beak_out variable (Overall; Gender; Age; Race)
- `break_out` - `character` - the group within the break_out_category that the percent_value corresponds to. (Overall; Male; Female; 25-44; 45-64; 65+; 35+; 75+; Non-Hispanic White; Non-Hispanic Black; Non-Hispanic Asian; Other)

```
Heart <- read.csv("heart.csv")
head(Heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  52   1  0      125  212   0       1     168     0     1.0     2  2    3
## 2  53   1  0      140  203   1       0     155     1     3.1     0  0    3
## 3  70   1  0      145  174   0       1     125     1     2.6     0  0    3
## 4  61   1  0      148  203   0       1     161     0     0.0     2  1    3
## 5  62   0  0      138  294   1       1     106     0     1.9     1  3    2
## 6  58   0  0      100  248   0       0     122     0     1.0     1  0    2
##   target
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
```

Simple Model

My main goal is to be able to predict the presents of heart disease in patients based on different risk factors. To do this I will use classification to determine if a patient belongs to the healthy heart group or the unhealthy group. I will use K-Nearest neighbors to do this classification.

Model Version 1:

```
heart <- as_tibble(Heart) %>%
  select(-target)

set.seed(131)

#Split training and test dataset
train_rows <- as.vector(createDataPartition(Heart$target, p = 0.8, list = FALSE))

train <- heart[train_rows, ]
test <- heart[-train_rows, ]
train.def <- Heart[train_rows, 14]
test.def <- Heart[-train_rows, 14]

knn.3 <- knn(train, test, train.def, k = 3)
knn.5 <- knn(train, test, train.def, k = 5)
knn.7 <- knn(train, test, train.def, k = 7)
knn.15 <- knn(train, test, train.def, k = 15)

#function that divides the correct predictions by total number of predictions that tell us how accurate
get.accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))}

tab.3 <- table(knn.3, test.def)
confusionMatrix(tab.3)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.3  0  1
##     0 74 26
##     1 19 86
```

```
##
##                Accuracy : 0.7805
##                  95% CI : (0.7175, 0.8352)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 2.543e-12
##
##                   Kappa : 0.56
##
##  Mcnemar's Test P-Value : 0.3711
##
##             Sensitivity : 0.7957
##             Specificity : 0.7679
##          Pos Pred Value : 0.7400
##          Neg Pred Value : 0.8190
##              Prevalence : 0.4537
##          Detection Rate : 0.3610
##    Detection Prevalence : 0.4878
##       Balanced Accuracy : 0.7818
##
##        'Positive' Class : 0
##
```

```
acc3 <- get.accuracy(tab.3)
#K = 3 correctly classifies 78.05% of the outcomes

tab.5 <- table(knn.5, test.def)
confusionMatrix(tab.5)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.5  0  1
##     0 70 31
##     1 23 81
##
##                Accuracy : 0.7366
##                  95% CI : (0.6707, 0.7955)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 1.558e-08
##
##                   Kappa : 0.4725
##
##  Mcnemar's Test P-Value : 0.3408
##
##             Sensitivity : 0.7527
##             Specificity : 0.7232
##          Pos Pred Value : 0.6931
##          Neg Pred Value : 0.7788
##              Prevalence : 0.4537
##          Detection Rate : 0.3415
##    Detection Prevalence : 0.4927
##       Balanced Accuracy : 0.7380
##
##        'Positive' Class : 0
```

```
##
```

```
acc5 <- get.accuracy(tab.5)
#K = 5 correctly classifies 73.66% of the outcomes

tab.7 <- table(knn.7, test.def)
confusionMatrix(tab.7)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.7  0  1
##     0 74 23
##     1 19 89
##
##                Accuracy : 0.7951
##                  95% CI : (0.7333, 0.8482)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 8.543e-14
##
##                   Kappa : 0.5882
##
##  Mcnemar's Test P-Value : 0.6434
##
##             Sensitivity : 0.7957
##             Specificity : 0.7946
##          Pos Pred Value : 0.7629
##          Neg Pred Value : 0.8241
##              Prevalence : 0.4537
##          Detection Rate : 0.3610
##    Detection Prevalence : 0.4732
##       Balanced Accuracy : 0.7952
##
##        'Positive' Class : 0
##
```

```
acc7 <- get.accuracy(tab.7)
#K = 7 correctly classifies 79.51% of the outcomes

tab.15 <- table(knn.15, test.def)
confusionMatrix(tab.15)
```

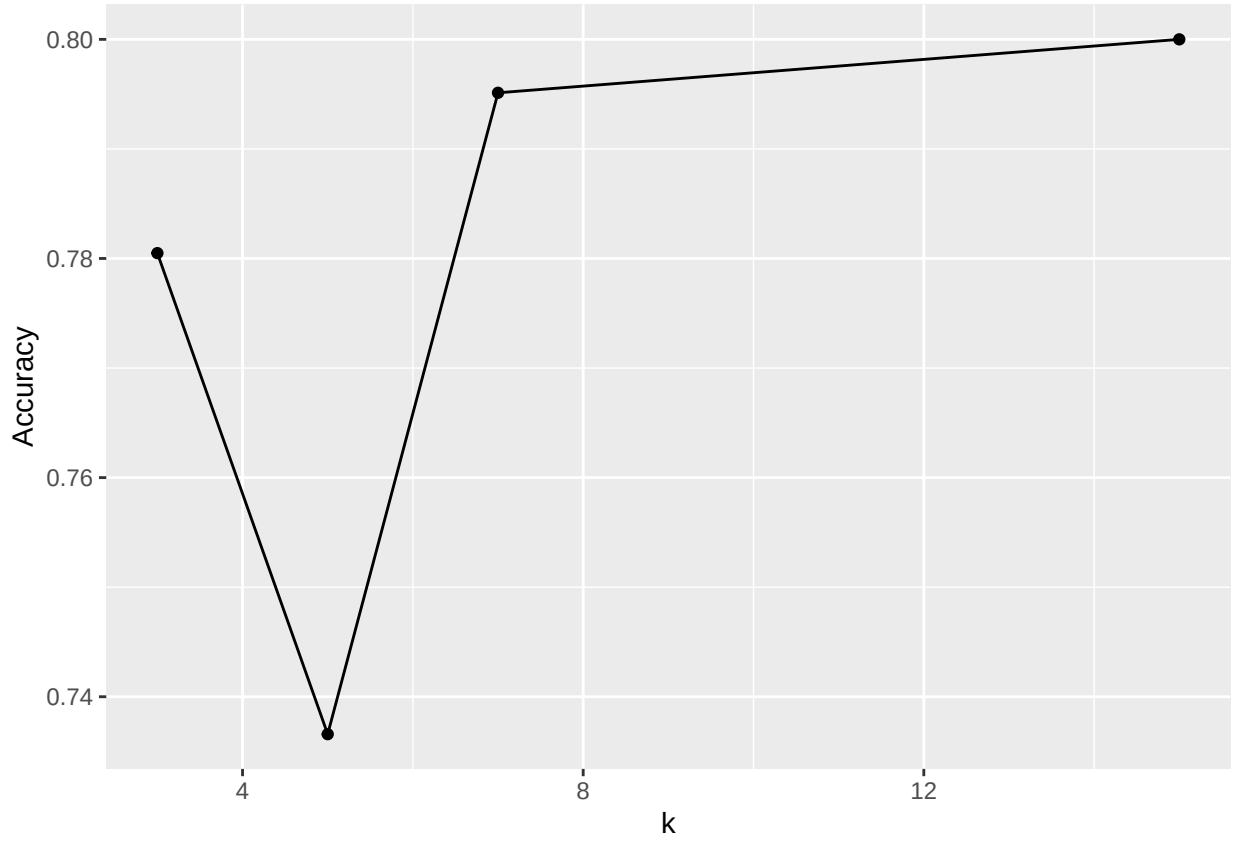
```
## Confusion Matrix and Statistics
##
##       test.def
## knn.15  0  1
##      0 75 23
##      1 18 89
##
##                Accuracy : 0.8
##                  95% CI : (0.7386, 0.8525)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 2.601e-14
```

```
##
##                   Kappa : 0.5984
##
##  Mcnemar's Test P-Value : 0.5322
##
##             Sensitivity : 0.8065
##             Specificity : 0.7946
##          Pos Pred Value : 0.7653
##          Neg Pred Value : 0.8318
##              Prevalence : 0.4537
##          Detection Rate : 0.3659
##    Detection Prevalence : 0.4780
##       Balanced Accuracy : 0.8005
##
##        'Positive' Class : 0
##
```

```
acc15 <- get.accuracy(tab.15)
#K = 15 correctly classifies 80% of the outcomes

k <- c(3, 5, 7, 15)
Accuracy <- c(acc3, acc5, acc7, acc15)
df <- data.frame(k, Accuracy)

#Accuracy plot
ggplot(df, aes(x=k, y=Accuracy)) +
  geom_point() +
  geom_line()
```



Model Version 2:

```
heart <- as_tibble(Heart) %>%
  select(-target, -ca, -trestbps)

set.seed(131)

#Split training and test dataset
train_rows <- as.vector(createDataPartition(Heart$target, p = 0.8, list = FALSE))

train <- heart[train_rows, ]
test <- heart[-train_rows, ]
train.def <- Heart[train_rows, 14]
test.def <- Heart[-train_rows, 14]

knn.3 <- knn(train, test, train.def, k = 3)
knn.5 <- knn(train, test, train.def, k = 5)
knn.7 <- knn(train, test, train.def, k = 7)
knn.15 <- knn(train, test, train.def, k = 15)

#function that divides the correct predictions by total number of predictions that tell us how accurate
get.accuracy <- function(x){sum(diag(x)/(sum(rowSums(x))))}

tab.3 <- table(knn.3, test.def)
confusionMatrix(tab.3)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.3  0  1
##     0 75 26
##     1 18 86
##
##                Accuracy : 0.7854
##                  95% CI : (0.7228, 0.8395)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 8.443e-13
##
##                   Kappa : 0.5701
##
##  Mcnemar's Test P-Value : 0.2913
##
##             Sensitivity : 0.8065
##             Specificity : 0.7679
##          Pos Pred Value : 0.7426
##          Neg Pred Value : 0.8269
##              Prevalence : 0.4537
##          Detection Rate : 0.3659
##    Detection Prevalence : 0.4927
##       Balanced Accuracy : 0.7872
##
##        'Positive' Class : 0
##
```

```
acc3 <- get.accuracy(tab.3)
#K = 3 correctly classifies 78.54% of the outcomes

tab.5 <- table(knn.5, test.def)
confusionMatrix(tab.5)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.5  0  1
##     0 77 30
##     1 16 82
##
##                Accuracy : 0.7756
##                  95% CI : (0.7123, 0.8308)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 7.449e-12
##
##                   Kappa : 0.553
##
##  Mcnemar's Test P-Value : 0.05527
##
##             Sensitivity : 0.8280
##             Specificity : 0.7321
##          Pos Pred Value : 0.7196
##          Neg Pred Value : 0.8367
```

```
##              Prevalence : 0.4537
##          Detection Rate : 0.3756
##    Detection Prevalence : 0.5220
##       Balanced Accuracy : 0.7800
##
##        'Positive' Class : 0
##
```

```
acc5 <- get.accuracy(tab.5)
#K = 5 correctly classifies 77.56% of the outcomes

tab.7 <- table(knn.7, test.def)
confusionMatrix(tab.7)
```

```
## Confusion Matrix and Statistics
##
##      test.def
## knn.7  0  1
##     0 75 20
##     1 18 92
##
##                Accuracy : 0.8146
##                  95% CI : (0.7546, 0.8654)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 6.125e-16
##
##                   Kappa : 0.6267
##
##  Mcnemar's Test P-Value : 0.8711
##
##             Sensitivity : 0.8065
##             Specificity : 0.8214
##          Pos Pred Value : 0.7895
##          Neg Pred Value : 0.8364
##              Prevalence : 0.4537
##          Detection Rate : 0.3659
##    Detection Prevalence : 0.4634
##       Balanced Accuracy : 0.8139
##
##        'Positive' Class : 0
##
```

```
acc7 <- get.accuracy(tab.7)
#K = 7 correctly classifies 81.46% of the outcomes

tab.15 <- table(knn.15, test.def)
confusionMatrix(tab.15)
```

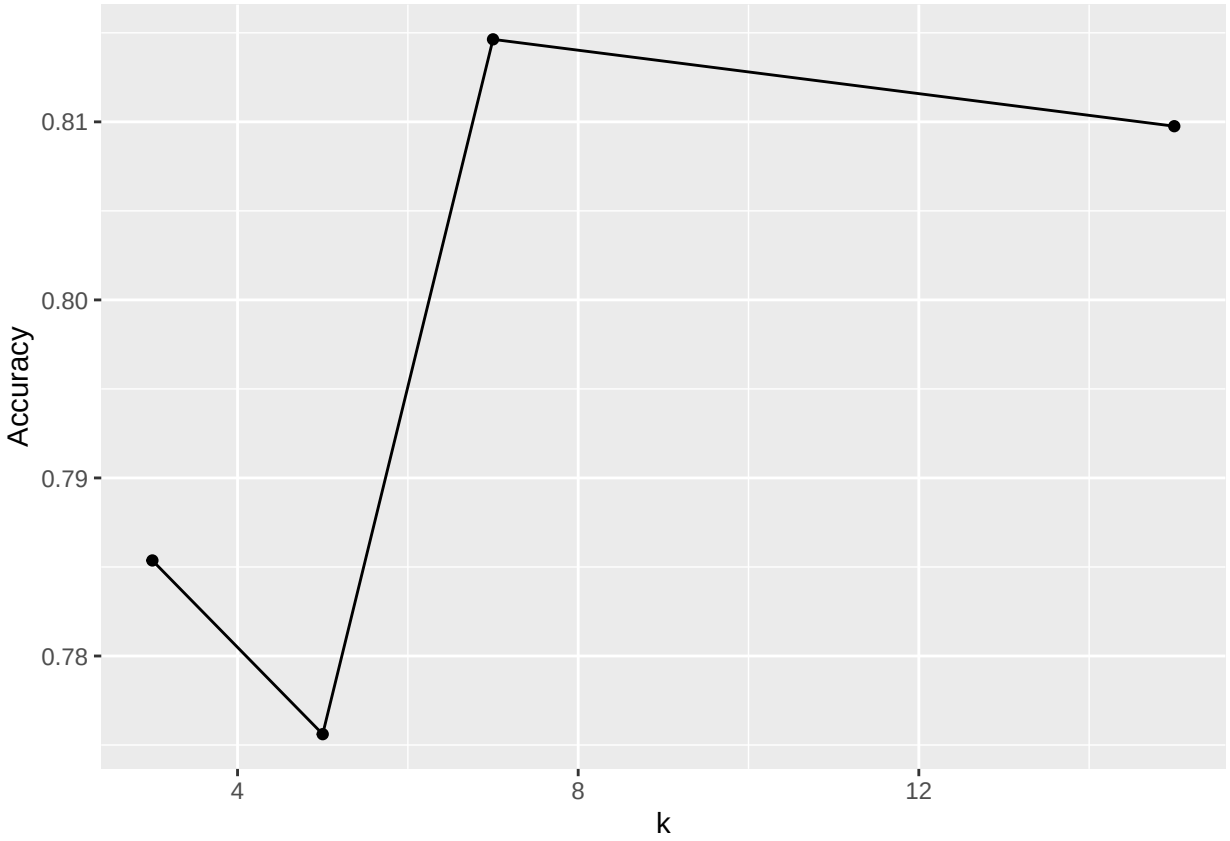
```
## Confusion Matrix and Statistics
##
##       test.def
## knn.15  0  1
##      0 78 24
```

```
##      1 15 88
##
##                Accuracy : 0.8098
##                  95% CI : (0.7492, 0.8611)
##     No Information Rate : 0.5463
##     P-Value [Acc > NIR] : 2.204e-15
##
##                   Kappa : 0.6193
##
##  Mcnemar's Test P-Value : 0.2002
##
##             Sensitivity : 0.8387
##             Specificity : 0.7857
##          Pos Pred Value : 0.7647
##          Neg Pred Value : 0.8544
##              Prevalence : 0.4537
##          Detection Rate : 0.3805
##    Detection Prevalence : 0.4976
##       Balanced Accuracy : 0.8122
##
##        'Positive' Class : 0
##
```

```
acc15 <- get.accuracy(tab.15)
#K = 15 correctly classifies 80.98% of the outcomes

k <- c(3, 5, 7, 15)
Accuracy <- c(acc3, acc5, acc7, acc15)
df <- data.frame(k, Accuracy)

#Accuracy plot
ggplot(df, aes(x=k, y=Accuracy)) +
  geom_point() +
  geom_line()
```



Parameters and Output

In the first version of my model, my parameters were all variables except the target and my best output was 80% accuracy. I choose to use all variables because the variable provided in the dataset are a collection of some of the best indicators of heart diseases, so it would make sense to use them all. The percent of success was a bit lower then I hoped for, so I decided to try another version where I removed some variables based on my data exploration from the first deliverable. For this version I removed the variables 'ca' and 'trestbps' from my parameters. The reason I removed 'trestbps' or resting blood pressure was because the distribution between the two groups in my first deliverable was very similar. This seemed to improve my model raising my best output to 81.46% accuracy. The biggest limitation to this model is my lack of knowledge and experience using k nearest neighbor, for example I am unsure how I could create a visualization for this algorithm.

Social and Ethical Implications

A positive social impact of this model is to reduce possible deaths on a global scale, since heart disease is the number one cause of death world wide. One possible ethical impact is the possibility of this model being used by companies such as private health insurance companies to determine whether or not to cover or raise prices on individuals who may be at risk of heart disease.