Predicting Parkinson's Based on Auditory Data

Kanika Chopra

2022-12-14

The scientific question of this project is to predict whether the patient has Parkinson's disease using patient's vocal features.

Import Data

First, we want to read in our data.

```
library(RCurl)
data <- read.csv(text = getURL(
  "https://raw.githubusercontent.com/kanikadchopra/Parkinsons-Prediction/main/parkinson_data.csv"))
```

This is a multivariate dataset with data from June 26, 2008.

```
attach(data)
```

Exploratory Data Analysis

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::complete() masks RCurl::complete()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
```

```
library(ggplot2)
```

Let's take a look at the dimensions of our data.

```
dim(data)
```

```
## [1] 195  24
```

We are working with 195 records with 24 attributes.

```
glimpse(data)
```

```
## Rows: 195
## Columns: 24
## $ name             <chr> "phon_R01_S01_1", "phon_R01_S01_2", "phon_R01_S01_3",~
## $ MDVP.Fo.Hz.      <dbl> 119.992, 122.400, 116.682, 116.676, 116.014, 120.552,~
## $ MDVP.Fhi.Hz.     <dbl> 157.302, 148.650, 131.111, 137.871, 141.781, 131.162,~
```

```
## $ MDVP.Flo.Hz.     <dbl> 74.997, 113.819, 111.555, 111.366, 110.655, 113.787, ~
## $ MDVP.Jitter...   <dbl> 0.00784, 0.00968, 0.01050, 0.00997, 0.01284, 0.00968,~
## $ MDVP.Jitter.Abs. <dbl> 0.00007, 0.00008, 0.00009, 0.00009, 0.00011, 0.00008,~
## $ MDVP.RAP         <dbl> 0.00370, 0.00465, 0.00544, 0.00502, 0.00655, 0.00463,~
## $ MDVP.PPQ         <dbl> 0.00554, 0.00696, 0.00781, 0.00698, 0.00908, 0.00750,~
## $ Jitter.DDP       <dbl> 0.01109, 0.01394, 0.01633, 0.01505, 0.01966, 0.01388,~
## $ MDVP.Shimmer     <dbl> 0.04374, 0.06134, 0.05233, 0.05492, 0.06425, 0.04701,~
## $ MDVP.Shimmer.dB. <dbl> 0.426, 0.626, 0.482, 0.517, 0.584, 0.456, 0.140, 0.13~
## $ Shimmer.APQ3     <dbl> 0.02182, 0.03134, 0.02757, 0.02924, 0.03490, 0.02328,~
## $ Shimmer.APQ5     <dbl> 0.03130, 0.04518, 0.03858, 0.04005, 0.04825, 0.03526,~
## $ MDVP.APQ         <dbl> 0.02971, 0.04368, 0.03590, 0.03772, 0.04465, 0.03243,~
## $ Shimmer.DDA      <dbl> 0.06545, 0.09403, 0.08270, 0.08771, 0.10470, 0.06985,~
## $ NHR              <dbl> 0.02211, 0.01929, 0.01309, 0.01353, 0.01767, 0.01222,~
## $ HNR              <dbl> 21.033, 19.085, 20.651, 20.644, 19.649, 21.378, 24.88~
## $ status           <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ RPDE             <dbl> 0.414783, 0.458359, 0.429895, 0.434969, 0.417356, 0.4~
## $ DFA              <dbl> 0.815285, 0.819521, 0.825288, 0.819235, 0.823484, 0.8~
## $ spread1          <dbl> -4.813031, -4.075192, -4.443179, -4.117501, -3.747787~
## $ spread2          <dbl> 0.266482, 0.335590, 0.311173, 0.334147, 0.234513, 0.2~
## $ D2               <dbl> 2.301442, 2.486855, 2.342259, 2.405554, 2.332180, 2.1~
## $ PPE              <dbl> 0.284654, 0.368674, 0.332634, 0.368975, 0.410335, 0.3~
```

Taking a glimpse at our data, we can see that one column corresponds to the ASCII subject name and recording number. Status is our response variable and is binary. The remainder of our values are all continuous variables.

Missing Data

Next, we want to conduct a few quick quality checks, such as checking if there are any missing values.

```
data %>% summarise_all(~ sum(is.na(.)))
```

```
##   name MDVP.Fo.Hz. MDVP.Fhi.Hz. MDVP.Flo.Hz. MDVP.Jitter... MDVP.Jitter.Abs.
## 1    0           0            0            0              0                0
##   MDVP.RAP MDVP.PPQ Jitter.DDP MDVP.Shimmer MDVP.Shimmer.dB. Shimmer.APQ3
## 1        0        0          0            0                0            0
##   Shimmer.APQ5 MDVP.APQ Shimmer.DDA NHR HNR status RPDE DFA spread1 spread2 D2
## 1            0        0           0   0   0      0    0   0       0       0  0
##   PPE
## 1   0
```

We can see that we have zero missing values along all of our columns which is good news as we don't have to handle them in the analysis.

Preliminary Plots

```
data %>%
  group_by(status) %>%
  count()
```

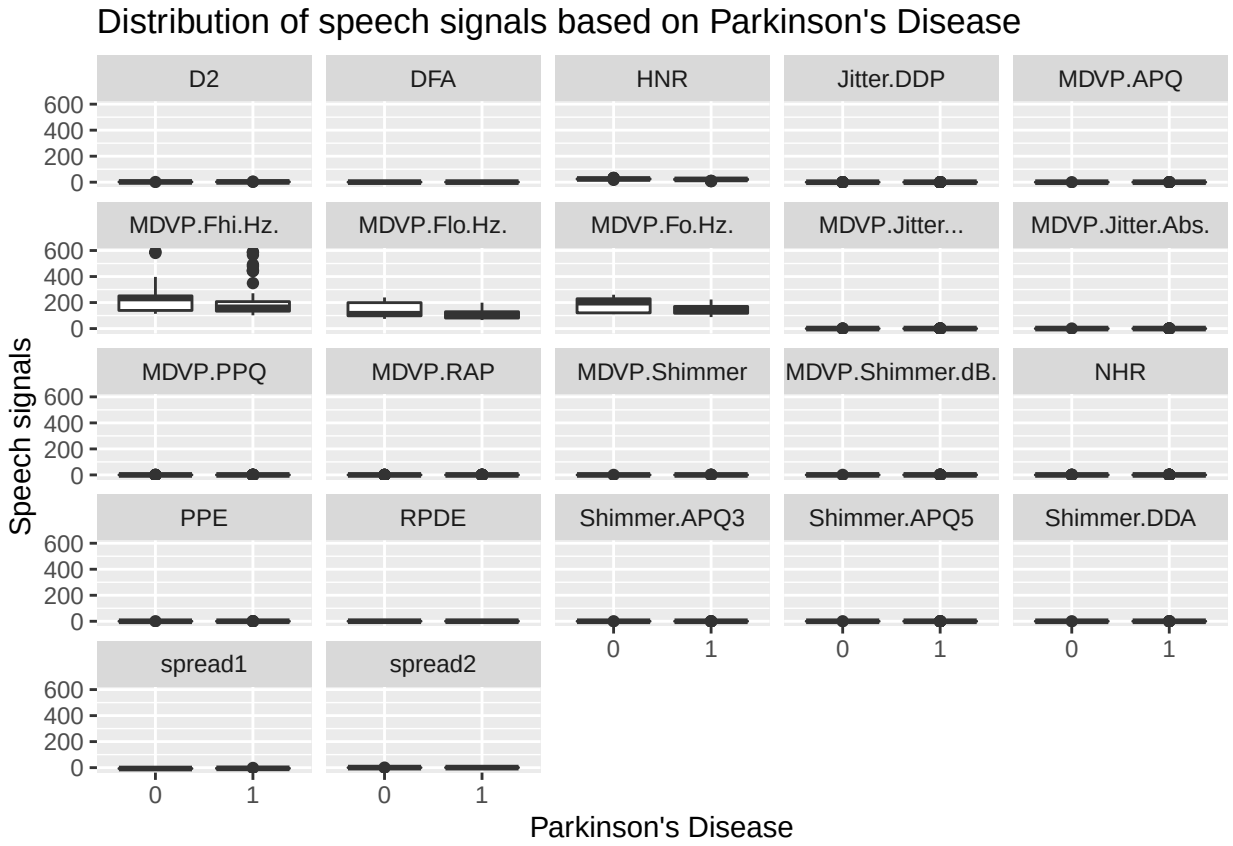
```
## # A tibble: 2 x 2
## # Groups:   status [2]
##   status     n
##    <int> <int>
## 1      0    48
## 2      1   147
```

Firstly, we can see that our data is skewed in that we have more data on patients with Parkinson's (`status=1`) than patients who do not have Parkinson's (`status=0`). We are working with imbalanced data in this case so that will be important to keep in mind. Let's plot this as a bar plot as well.

```
data$status <- factor(data$status)
data %>%
  group_by(status) %>%
  count() %>%
  ggplot(., aes(x=status, y=n, fill=n)) +
  geom_bar(stat="identity", fill='#4169E1') +
  ggtitle("Distribution of patients with and without Parkinson's") +
  labs(y="Parkinson's Disease", x="Number of patients")
```

We'll explore the boxplots of our variables in comparison to status.

```
data %>%
  dplyr::select(-contains("name")) %>%
  gather(-status, key='var', value='value') %>%
  ggplot(aes(x=status, y=value, outlier.color='red')) +
  geom_boxplot() +
  facet_wrap(~ var) +
  labs(title = "Distribution of speech signals based on Parkinson's Disease",
       x = "Parkinson's Disease",
       y = "Speech signals")
```



From our plot we can see that our measurements are higher for Parkinson's disease for most variables except for HNR and MDVP.Hz (Fhi, Flo, Fo). In general, it seems that those with Parkinson's disease have a higher speech measurement. We can also see some potential outliers in the data; however, given that our data set is small, we do not remove these values as these may not be outliers if more data was provided.

Data Cleaning and Transformation

```
library(dplyr)
```

We can start by dropping the name column from our dataframe.

```
data <- data %>%
  dplyr::select(-contains("name"))
```

Next, we check the correlation between the variables.

```
library(corrr)
```

```
data %>% correlate()
```

```
## Non-numeric variables removed from input: `status`
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

```
## # A tibble: 22 x 23
##    term         MDVP.~1 MDVP.F~2 MDVP.~3 MDVP.~4 MDVP.~5 MDVP.~6 MDVP.~7 Jitte~8
##    <chr>          <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 MDVP.Fo.Hz.   NA       0.401    0.597   -0.118 -0.382  -0.0762 -0.112  -0.0762
```

```
##  2 MDVP.Fhi.Hz.  0.401  NA        0.0850   0.102 -0.0292  0.0972  0.0911  0.0971
##  3 MDVP.Flo.Hz.  0.597   0.0850  NA       -0.140 -0.278  -0.101  -0.0958 -0.100
##  4 MDVP.Jitter~ -0.118   0.102   -0.140   NA      0.936   0.990   0.974   0.990
##  5 MDVP.Jitter~ -0.382  -0.0292  -0.278    0.936 NA       0.923   0.898   0.923
##  6 MDVP.RAP     -0.0762  0.0972  -0.101    0.990  0.923  NA       0.957   1.00
##  7 MDVP.PPQ     -0.112   0.0911  -0.0958   0.974  0.898   0.957  NA       0.957
##  8 Jitter.DDP   -0.0762  0.0971  -0.100    0.990  0.923   1.00    0.957  NA
##  9 MDVP.Shimmer -0.0984  0.00228 -0.145    0.769  0.703   0.760   0.798   0.760
## 10 MDVP.Shimme~ -0.0737  0.0435  -0.119    0.804  0.717   0.791   0.839   0.791
## # ... with 12 more rows, 14 more variables: MDVP.Shimmer <dbl>,
## #   MDVP.Shimmer.dB. <dbl>, Shimmer.APQ3 <dbl>, Shimmer.APQ5 <dbl>,
## #   MDVP.APQ <dbl>, Shimmer.DDA <dbl>, NHR <dbl>, HNR <dbl>, RPDE <dbl>,
## #   DFA <dbl>, spread1 <dbl>, spread2 <dbl>, D2 <dbl>, PPE <dbl>, and
## #   abbreviated variable names 1: MDVP.Fo.Hz., 2: MDVP.Fhi.Hz.,
## #   3: MDVP.Flo.Hz., 4: MDVP.Jitter..., 5: MDVP.Jitter.Abs., 6: MDVP.RAP,
## #   7: MDVP.PPQ, 8: Jitter.DDP
```

This is harder to see numerically so we plot a heatmap of the correlation values. We take the absolute value of the correlation plot to make it easier to notice which variables are correlated with one another.

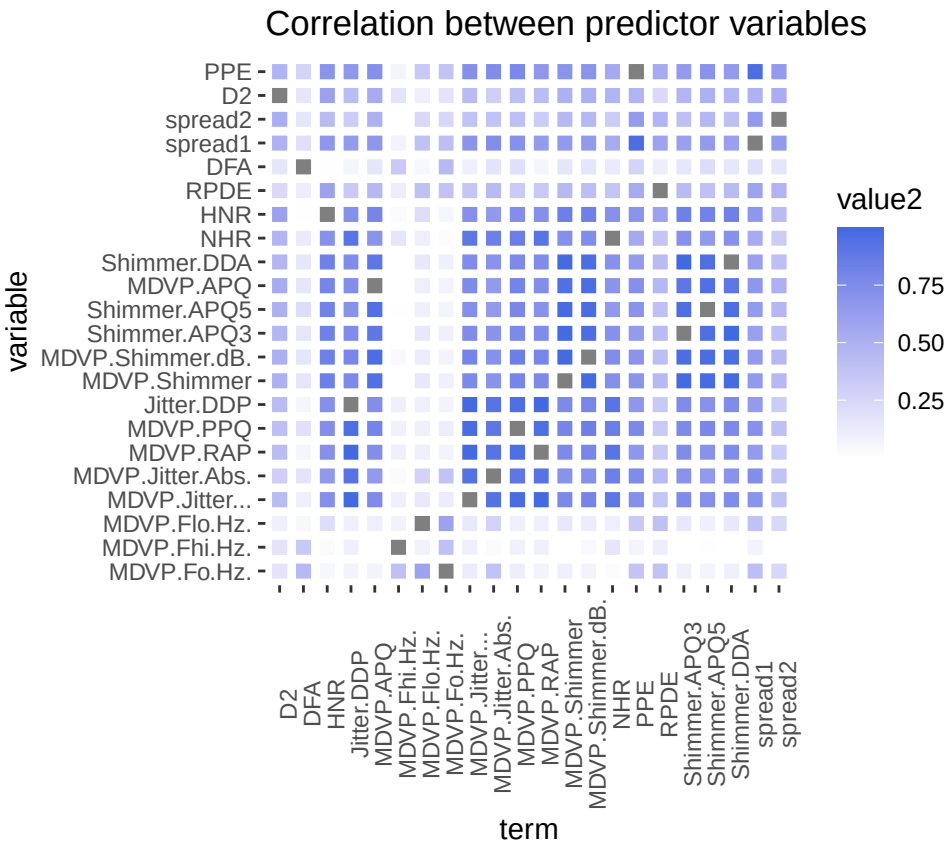
```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
data %>%
  correlate() %>%
  melt(.) %>%
  mutate(
    value2 = abs(value)) %>%
  ggplot(., aes(x=term, y=variable, fill=value2)) +
  geom_tile(color = "white",
            lwd = 1.5,
            linetype = 1) + theme(axis.text.x = element_text(angle = 90)) +
  scale_fill_gradient(low = "white", high = "#4169E1") + coord_fixed() +
  ggtitle('Correlation between predictor variables')
```

```
## Non-numeric variables removed from input: `status`
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
## Using term as id variables
```



We can see a lot of correlated variables. We note that `status` does not have a high correlation with any of the values. To get a subset of variables that we should drop due to multicollinearity issues, we take a look at a subset of our data, i.e. correlations that are greater than 0.5 in absolute value.

```
correlations = melt(correlate(data))
```

```
## Non-numeric variables removed from input: `status`
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
## Using term as id variables
```

```
correlations$value <- abs(correlations$value)

correlations[correlations$value > 0.6, ]
```

```
##                   term         variable     value
## NA                <NA>             <NA>        NA
## NA.1              <NA>             <NA>        NA
## NA.2              <NA>             <NA>        NA
## NA.3              <NA>             <NA>        NA
## 71    MDVP.Jitter.Abs.   MDVP.Jitter... 0.9357140
## 72            MDVP.RAP   MDVP.Jitter... 0.9902756
## 73            MDVP.PPQ   MDVP.Jitter... 0.9742564
## 74          Jitter.DDP   MDVP.Jitter... 0.9902762
## 75        MDVP.Shimmer   MDVP.Jitter... 0.7690632
## 76    MDVP.Shimmer.dB.   MDVP.Jitter... 0.8042893
## 77        Shimmer.APQ3   MDVP.Jitter... 0.7466252
## 78        Shimmer.APQ5   MDVP.Jitter... 0.7255610
```

```
## 79            MDVP.APQ   MDVP.Jitter... 0.7582553
## 80         Shimmer.DDA   MDVP.Jitter... 0.7466352
## 81                 NHR   MDVP.Jitter... 0.9069586
## 82                 HNR   MDVP.Jitter... 0.7281651
## 85             spread1   MDVP.Jitter... 0.6935767
## 88                 PPE   MDVP.Jitter... 0.7215429
## 92      MDVP.Jitter... MDVP.Jitter.Abs. 0.9357140
## NA.4             <NA>             <NA>        NA
## 94            MDVP.RAP MDVP.Jitter.Abs. 0.9229110
## 95            MDVP.PPQ MDVP.Jitter.Abs. 0.8977779
## 96          Jitter.DDP MDVP.Jitter.Abs. 0.9229130
## 97        MDVP.Shimmer MDVP.Jitter.Abs. 0.7033224
## 98    MDVP.Shimmer.dB. MDVP.Jitter.Abs. 0.7166013
## 99        Shimmer.APQ3 MDVP.Jitter.Abs. 0.6971530
## 100       Shimmer.APQ5 MDVP.Jitter.Abs. 0.6489607
## 101           MDVP.APQ MDVP.Jitter.Abs. 0.6487934
## 102        Shimmer.DDA MDVP.Jitter.Abs. 0.6971697
## 103                NHR MDVP.Jitter.Abs. 0.8349722
## 104                HNR MDVP.Jitter.Abs. 0.6568096
## 107            spread1 MDVP.Jitter.Abs. 0.7357792
## 110                PPE MDVP.Jitter.Abs. 0.7481617
## 114     MDVP.Jitter...         MDVP.RAP 0.9902756
## 115   MDVP.Jitter.Abs.         MDVP.RAP 0.9229110
## NA.5             <NA>             <NA>        NA
## 117           MDVP.PPQ         MDVP.RAP 0.9573169
## 118         Jitter.DDP         MDVP.RAP 0.9999996
## 119       MDVP.Shimmer         MDVP.RAP 0.7595805
## 120   MDVP.Shimmer.dB.         MDVP.RAP 0.7906515
## 121       Shimmer.APQ3         MDVP.RAP 0.7449124
## 122       Shimmer.APQ5         MDVP.RAP 0.7099268
## 123           MDVP.APQ         MDVP.RAP 0.7374547
## 124        Shimmer.DDA         MDVP.RAP 0.7449192
## 125                NHR         MDVP.RAP 0.9195207
## 126                HNR         MDVP.RAP 0.7215432
## 129            spread1         MDVP.RAP 0.6483278
## 132                PPE         MDVP.RAP 0.6709990
## 136     MDVP.Jitter...         MDVP.PPQ 0.9742564
## 137   MDVP.Jitter.Abs.         MDVP.PPQ 0.8977779
## 138           MDVP.RAP         MDVP.PPQ 0.9573169
## NA.6             <NA>             <NA>        NA
## 140         Jitter.DDP         MDVP.PPQ 0.9573192
## 141       MDVP.Shimmer         MDVP.PPQ 0.7978260
## 142   MDVP.Shimmer.dB.         MDVP.PPQ 0.8392389
## 143       Shimmer.APQ3         MDVP.PPQ 0.7635799
## 144       Shimmer.APQ5         MDVP.PPQ 0.7867805
## 145           MDVP.APQ         MDVP.PPQ 0.8041393
## 146        Shimmer.DDA         MDVP.PPQ 0.7635922
## 147                NHR         MDVP.PPQ 0.8446035
## 148                HNR         MDVP.PPQ 0.7315105
## 151            spread1         MDVP.PPQ 0.7164886
## 154                PPE         MDVP.PPQ 0.7696473
## 158     MDVP.Jitter...       Jitter.DDP 0.9902762
## 159   MDVP.Jitter.Abs.       Jitter.DDP 0.9229130
## 160           MDVP.RAP       Jitter.DDP 0.9999996
```

```
## 161           MDVP.PPQ       Jitter.DDP 0.9573192
## NA.7              <NA>             <NA>        NA
## 163       MDVP.Shimmer       Jitter.DDP 0.7595547
## 164   MDVP.Shimmer.dB.       Jitter.DDP 0.7906206
## 165       Shimmer.APQ3       Jitter.DDP 0.7448938
## 166       Shimmer.APQ5       Jitter.DDP 0.7099071
## 167           MDVP.APQ       Jitter.DDP 0.7374387
## 168        Shimmer.DDA       Jitter.DDP 0.7449006
## 169                NHR       Jitter.DDP 0.9195482
## 170                HNR       Jitter.DDP 0.7214944
## 173            spread1       Jitter.DDP 0.6483276
## 176                PPE       Jitter.DDP 0.6710053
## 180     MDVP.Jitter...     MDVP.Shimmer 0.7690632
## 181   MDVP.Jitter.Abs.     MDVP.Shimmer 0.7033224
## 182           MDVP.RAP     MDVP.Shimmer 0.7595805
## 183           MDVP.PPQ     MDVP.Shimmer 0.7978260
## 184         Jitter.DDP     MDVP.Shimmer 0.7595547
## NA.8              <NA>             <NA>        NA
## 186   MDVP.Shimmer.dB.     MDVP.Shimmer 0.9872578
## 187       Shimmer.APQ3     MDVP.Shimmer 0.9876251
## 188       Shimmer.APQ5     MDVP.Shimmer 0.9828354
## 189           MDVP.APQ     MDVP.Shimmer 0.9500829
## 190        Shimmer.DDA     MDVP.Shimmer 0.9876257
## 191                NHR     MDVP.Shimmer 0.7221945
## 192                HNR     MDVP.Shimmer 0.8352707
## 195            spread1     MDVP.Shimmer 0.6547343
## 198                PPE     MDVP.Shimmer 0.6937707
## 202     MDVP.Jitter... MDVP.Shimmer.dB. 0.8042893
## 203   MDVP.Jitter.Abs. MDVP.Shimmer.dB. 0.7166013
## 204           MDVP.RAP MDVP.Shimmer.dB. 0.7906515
## 205           MDVP.PPQ MDVP.Shimmer.dB. 0.8392389
## 206         Jitter.DDP MDVP.Shimmer.dB. 0.7906206
## 207       MDVP.Shimmer MDVP.Shimmer.dB. 0.9872578
## NA.9              <NA>             <NA>        NA
## 209       Shimmer.APQ3 MDVP.Shimmer.dB. 0.9631981
## 210       Shimmer.APQ5 MDVP.Shimmer.dB. 0.9737506
## 211           MDVP.APQ MDVP.Shimmer.dB. 0.9609767
## 212        Shimmer.DDA MDVP.Shimmer.dB. 0.9632017
## 213                NHR MDVP.Shimmer.dB. 0.7444773
## 214                HNR MDVP.Shimmer.dB. 0.8278053
## 217            spread1 MDVP.Shimmer.dB. 0.6525467
## 220                PPE MDVP.Shimmer.dB. 0.6950581
## 224     MDVP.Jitter...     Shimmer.APQ3 0.7466252
## 225   MDVP.Jitter.Abs.     Shimmer.APQ3 0.6971530
## 226           MDVP.RAP     Shimmer.APQ3 0.7449124
## 227           MDVP.PPQ     Shimmer.APQ3 0.7635799
## 228         Jitter.DDP     Shimmer.APQ3 0.7448938
## 229       MDVP.Shimmer     Shimmer.APQ3 0.9876251
## 230   MDVP.Shimmer.dB.     Shimmer.APQ3 0.9631981
## NA.10             <NA>             <NA>        NA
## 232       Shimmer.APQ5     Shimmer.APQ3 0.9600698
## 233           MDVP.APQ     Shimmer.APQ3 0.8966445
## 234        Shimmer.DDA     Shimmer.APQ3 1.0000000
## 235                NHR     Shimmer.APQ3 0.7162067
```

```
## 236               HNR     Shimmer.APQ3 0.8271233
## 239           spread1     Shimmer.APQ3 0.6109674
## 242               PPE     Shimmer.APQ3 0.6453767
## 246     MDVP.Jitter...     Shimmer.APQ5 0.7255610
## 247   MDVP.Jitter.Abs.     Shimmer.APQ5 0.6489607
## 248          MDVP.RAP     Shimmer.APQ5 0.7099268
## 249          MDVP.PPQ     Shimmer.APQ5 0.7867805
## 250        Jitter.DDP     Shimmer.APQ5 0.7099071
## 251       MDVP.Shimmer     Shimmer.APQ5 0.9828354
## 252   MDVP.Shimmer.dB.     Shimmer.APQ5 0.9737506
## 253       Shimmer.APQ3     Shimmer.APQ5 0.9600698
## NA.11             <NA>             <NA>        NA
## 255          MDVP.APQ     Shimmer.APQ5 0.9491461
## 256        Shimmer.DDA     Shimmer.APQ5 0.9600716
## 257               NHR     Shimmer.APQ5 0.6580798
## 258               HNR     Shimmer.APQ5 0.8137528
## 261           spread1     Shimmer.APQ5 0.6468089
## 264               PPE     Shimmer.APQ5 0.7024557
## 268     MDVP.Jitter...         MDVP.APQ 0.7582553
## 269   MDVP.Jitter.Abs.         MDVP.APQ 0.6487934
## 270          MDVP.RAP         MDVP.APQ 0.7374547
## 271          MDVP.PPQ         MDVP.APQ 0.8041393
## 272        Jitter.DDP         MDVP.APQ 0.7374387
## 273       MDVP.Shimmer         MDVP.APQ 0.9500829
## 274   MDVP.Shimmer.dB.         MDVP.APQ 0.9609767
## 275       Shimmer.APQ3         MDVP.APQ 0.8966445
## 276       Shimmer.APQ5         MDVP.APQ 0.9491461
## NA.12             <NA>             <NA>        NA
## 278        Shimmer.DDA         MDVP.APQ 0.8966468
## 279               NHR         MDVP.APQ 0.6940190
## 280               HNR         MDVP.APQ 0.8004066
## 283           spread1         MDVP.APQ 0.6731581
## 286               PPE         MDVP.APQ 0.7216940
## 290     MDVP.Jitter...      Shimmer.DDA 0.7466352
## 291   MDVP.Jitter.Abs.      Shimmer.DDA 0.6971697
## 292          MDVP.RAP      Shimmer.DDA 0.7449192
## 293          MDVP.PPQ      Shimmer.DDA 0.7635922
## 294        Jitter.DDP      Shimmer.DDA 0.7449006
## 295       MDVP.Shimmer      Shimmer.DDA 0.9876257
## 296   MDVP.Shimmer.dB.      Shimmer.DDA 0.9632017
## 297       Shimmer.APQ3      Shimmer.DDA 1.0000000
## 298       Shimmer.APQ5      Shimmer.DDA 0.9600716
## 299           MDVP.APQ      Shimmer.DDA 0.8966468
## NA.13             <NA>             <NA>        NA
## 301               NHR      Shimmer.DDA 0.7162145
## 302               HNR      Shimmer.DDA 0.8271302
## 305           spread1      Shimmer.DDA 0.6109712
## 308               PPE      Shimmer.DDA 0.6453890
## 312     MDVP.Jitter...              NHR 0.9069586
## 313   MDVP.Jitter.Abs.              NHR 0.8349722
## 314          MDVP.RAP              NHR 0.9195207
## 315          MDVP.PPQ              NHR 0.8446035
## 316        Jitter.DDP              NHR 0.9195482
## 317       MDVP.Shimmer              NHR 0.7221945
```

```
## 318   MDVP.Shimmer.dB.             NHR 0.7444773
## 319       Shimmer.APQ3             NHR 0.7162067
## 320       Shimmer.APQ5             NHR 0.6580798
## 321           MDVP.APQ             NHR 0.6940190
## 322        Shimmer.DDA             NHR 0.7162145
## NA.14             <NA>            <NA>        NA
## 324               HNR             NHR 0.7140724
## 334     MDVP.Jitter...             HNR 0.7281651
## 335   MDVP.Jitter.Abs.             HNR 0.6568096
## 336           MDVP.RAP             HNR 0.7215432
## 337           MDVP.PPQ             HNR 0.7315105
## 338         Jitter.DDP             HNR 0.7214944
## 339       MDVP.Shimmer             HNR 0.8352707
## 340   MDVP.Shimmer.dB.             HNR 0.8278053
## 341       Shimmer.APQ3             HNR 0.8271233
## 342       Shimmer.APQ5             HNR 0.8137528
## 343           MDVP.APQ             HNR 0.8004066
## 344        Shimmer.DDA             HNR 0.8271302
## 345               NHR             HNR 0.7140724
## NA.15             <NA>            <NA>        NA
## 349            spread1             HNR 0.6732098
## 351                D2             HNR 0.6014010
## 352               PPE             HNR 0.6928759
## NA.16             <NA>            <NA>        NA
## NA.17             <NA>            <NA>        NA
## 400     MDVP.Jitter...          spread1 0.6935767
## 401   MDVP.Jitter.Abs.          spread1 0.7357792
## 402           MDVP.RAP          spread1 0.6483278
## 403           MDVP.PPQ          spread1 0.7164886
## 404         Jitter.DDP          spread1 0.6483276
## 405       MDVP.Shimmer          spread1 0.6547343
## 406   MDVP.Shimmer.dB.          spread1 0.6525467
## 407       Shimmer.APQ3          spread1 0.6109674
## 408       Shimmer.APQ5          spread1 0.6468089
## 409           MDVP.APQ          spread1 0.6731581
## 410        Shimmer.DDA          spread1 0.6109712
## 412               HNR          spread1 0.6732098
## NA.18             <NA>             <NA>        NA
## 416            spread2          spread1 0.6523578
## 418               PPE          spread1 0.9624353
## 437            spread1          spread2 0.6523578
## NA.19             <NA>             <NA>        NA
## 440               PPE          spread2 0.6447110
## 456               HNR               D2 0.6014010
## NA.20             <NA>             <NA>        NA
## 466     MDVP.Jitter...              PPE 0.7215429
## 467   MDVP.Jitter.Abs.              PPE 0.7481617
## 468           MDVP.RAP              PPE 0.6709990
## 469           MDVP.PPQ              PPE 0.7696473
## 470         Jitter.DDP              PPE 0.6710053
## 471       MDVP.Shimmer              PPE 0.6937707
## 472   MDVP.Shimmer.dB.              PPE 0.6950581
## 473       Shimmer.APQ3              PPE 0.6453767
## 474       Shimmer.APQ5              PPE 0.7024557
```

```
## 475           MDVP.APQ             PPE 0.7216940
## 476       Shimmer.DDA             PPE 0.6453890
## 478               HNR             PPE 0.6928759
## 481           spread1             PPE 0.9624353
## 482           spread2             PPE 0.6447110
## NA.21            <NA>            <NA>        NA
```

We note that the following variables have high correlations with multiple other variables:

- MDVP.Jitter...
- MDVP.Jitter.Abs
- MDVP.RAP
- MDVP.PPQ
- Jitter.DDP
- MDVP.Shimmer
- MDVP.Shimmer.db
- Shimmer.APQ3
- Shimmer.APQ5
- MDVP.APQ
- Shimmer.DDA
- NHR
- HNR
- spread1
- PPE

Note that Jitter and Shimmer variables are highly correlated. This leaves us with 7 variables to work with which are (MDVP.Fo.Hz, MDVP.Fhi.Hz, MVDP.Flo.Hz, RPDE, DFA, spread2, D2) and then our status variable

This is important because to fit a logistic regression model, we are assuming there is no multicollinearity otherwise we would have high errors with the predictors. Next, we notice that we have a differing ranges of values for each category so we normalize our variables so that scale does not influence our prediction.

```
keep_variables <- c("MDVP.Fo.Hz.","MDVP.Fhi.Hz.", "MDVP.Flo.Hz.", "RPDE", "DFA", "spread2", "D2")
```

```
x <- data %>%
  dplyr::select(all_of(keep_variables)) %>%
  scale(.)

x <- data.frame(x)
y <- data$status
```

```
summary(x)
```

```
##   MDVP.Fo.Hz.       MDVP.Fhi.Hz.      MDVP.Flo.Hz.          RPDE
##  Min.   :-1.5921   Min.   :-1.0379   Min.   :-1.1684   Min.   :-2.32790
##  1st Qu.:-0.8856   1st Qu.:-0.6803   1st Qu.:-0.7360   1st Qu.:-0.74301
##  Median :-0.1314   Median :-0.2325   Median :-0.2759   Median :-0.02484
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.6895   3rd Qu.: 0.2962   3rd Qu.: 0.5444   3rd Qu.: 0.85651
##  Max.   : 2.5580   Max.   : 4.3165   Max.   : 2.8226   Max.   : 1.79539
##       DFA              spread2               D2
##  Min.   :-2.59899   Min.   :-2.64054   Min.   :-2.50403
##  1st Qu.:-0.78325   1st Qu.:-0.62537   1st Qu.:-0.73851
##  Median : 0.07509   Median :-0.09142   Median :-0.05301
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.79121   3rd Qu.: 0.63213   3rd Qu.: 0.66518
```

```
##  Max.   : 1.93706   Max.   : 2.68546   Max.   : 3.36816
```

We can now see that we have a mean of 0 across all of our variables.

Thus, we are now ready to build our logistic regression model.

Model Building

We want to first split our data into training and testing sets.

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(225)
idx <- x$spread2 %>% createDataPartition(p=0.8, list=FALSE)

train.data <- x[idx, ]
test.data <- x[-idx,]

train.y <- y[idx]
test.y <- y[-idx]

train <- data.frame(train.data, status=train.y)
```

It is important to note that we do not have a very large sample size but are working with 159 variables in our training data set. Hence, increasing our sample size would aid with fitting a more reliable model but that is a limitation with the scope of this project.

To build our model, we will use multiple methods to choose our best model. The first being LRT with `drop1`. Then, we will look at the significant factors with our full model and use stepAIC for alternative methods.

LRT

For this method, we'll include interaction effects to see if they are significant. We only include interaction terms of interest, i.e. the interaction between `MDVP.Fo.Hz.` and `MDVP.Fhi.Hz.` and `MDVP.Flo.Hz.` would not be relevant. We choose to use `MDVP.Fo.Hz.` to include with the interaction terms with the other variables.

```
inter.model <- glm(status ~ . + (MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2)^2, data=train,
                   family=binomial)
```

```
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + (MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2)^2
##                     Df Deviance    AIC    LRT Pr(>Chi)
## <none>                   78.224 114.22
## MDVP.Fhi.Hz.         1   79.480 113.48 1.2556  0.26249
## MDVP.Flo.Hz.         1   78.251 112.25 0.0265  0.87071
```

```
## MDVP.Fo.Hz.:RPDE     1   78.975 112.97 0.7511  0.38612
## MDVP.Fo.Hz.:DFA      1   79.358 113.36 1.1343  0.28686
## MDVP.Fo.Hz.:spread2  1   78.244 112.24 0.0198  0.88809
## MDVP.Fo.Hz.:D2       1   80.383 114.38 2.1591  0.14173
## RPDE:DFA             1   84.517 118.52 6.2929  0.01212 *
## RPDE:spread2         1   78.235 112.23 0.0108  0.91738
## RPDE:D2              1   79.338 113.34 1.1138  0.29126
## DFA:spread2          1   80.102 114.10 1.8775  0.17061
## DFA:D2               1   78.762 112.76 0.5382  0.46318
## spread2:D2           1   78.381 112.38 0.1567  0.69223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see in the first iteration that none of the interaction terms are significant except for `RPDE:DFA`. We drop the interaction term for `RPDE:spread1` since it has the largest p-value.

```
inter.model <- update(inter.model, .~. - RPDE:DFA)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:RPDE + MDVP.Fo.Hz.:DFA + MDVP.Fo.Hz.:spread2 +
##     MDVP.Fo.Hz.:D2 + RPDE:spread2 + RPDE:D2 + DFA:spread2 + DFA:D2 +
##     spread2:D2
##                      Df Deviance    AIC    LRT Pr(>Chi)
## <none>                    84.517 118.52
## MDVP.Fhi.Hz.          1   86.026 118.03 1.5088  0.21932
## MDVP.Flo.Hz.          1   85.504 117.50 0.9871  0.32046
## MDVP.Fo.Hz.:RPDE      1   84.837 116.84 0.3202  0.57151
## MDVP.Fo.Hz.:DFA       1   84.910 116.91 0.3928  0.53083
## MDVP.Fo.Hz.:spread2   1   84.606 116.61 0.0888  0.76577
## MDVP.Fo.Hz.:D2        1   88.210 120.21 3.6927  0.05465 .
## RPDE:spread2          1   84.521 116.52 0.0035  0.95252
## RPDE:D2               1   87.641 119.64 3.1244  0.07713 .
## DFA:spread2           1   85.255 117.25 0.7377  0.39041
## DFA:D2                1   85.714 117.71 1.1973  0.27386
## spread2:D2            1   85.592 117.59 1.0747  0.29989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we move remove the interaction effect for `RPDE:spread2`.

```
inter.model <- update(inter.model, .~. - RPDE:spread2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:RPDE + MDVP.Fo.Hz.:DFA + MDVP.Fo.Hz.:spread2 +
##     MDVP.Fo.Hz.:D2 + RPDE:D2 + DFA:spread2 + DFA:D2 + spread2:D2
##                      Df Deviance    AIC    LRT Pr(>Chi)
## <none>                    84.521 116.52
## MDVP.Fhi.Hz.          1   86.028 116.03 1.5071  0.21958
```

```
## MDVP.Flo.Hz.          1   85.565 115.56 1.0445  0.30678
## MDVP.Fo.Hz.:RPDE      1   84.838 114.84 0.3171  0.57337
## MDVP.Fo.Hz.:DFA       1   84.911 114.91 0.3900  0.53232
## MDVP.Fo.Hz.:spread2   1   84.619 114.62 0.0986  0.75346
## MDVP.Fo.Hz.:D2        1   88.302 118.30 3.7814  0.05182 .
## RPDE:D2               1   87.983 117.98 3.4620  0.06280 .
## DFA:spread2           1   85.523 115.52 1.0023  0.31675
## DFA:D2                1   85.758 115.76 1.2370  0.26606
## spread2:D2            1   85.598 115.60 1.0773  0.29931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we move remove the interaction effect for `MDVP.Fo.Hz.:spread2`.

```
inter.model <- update(inter.model, .~. - MDVP.Fo.Hz.:spread2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:RPDE + MDVP.Fo.Hz.:DFA + MDVP.Fo.Hz.:D2 +
##     RPDE:D2 + DFA:spread2 + DFA:D2 + spread2:D2
##                   Df Deviance    AIC    LRT Pr(>Chi)
## <none>                 84.619 114.62
## MDVP.Fhi.Hz.       1   86.036 114.04 1.4171  0.23389
## MDVP.Flo.Hz.       1   85.594 113.59 0.9745  0.32355
## MDVP.Fo.Hz.:RPDE   1   85.060 113.06 0.4407  0.50678
## MDVP.Fo.Hz.:DFA    1   85.026 113.03 0.4063  0.52387
## MDVP.Fo.Hz.:D2     1   88.309 116.31 3.6896  0.05475 .
## RPDE:D2            1   88.075 116.08 3.4553  0.06305 .
## DFA:spread2        1   85.928 113.93 1.3085  0.25266
## DFA:D2             1   85.852 113.85 1.2323  0.26696
## spread2:D2         1   86.585 114.58 1.9656  0.16091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we move remove the interaction effect for `MDVP.Fo.Hz.:DFA`.

```
inter.model <- update(inter.model, .~. - MDVP.Fo.Hz.:DFA)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:RPDE + MDVP.Fo.Hz.:D2 + RPDE:D2 +
##     DFA:spread2 + DFA:D2 + spread2:D2
##                   Df Deviance    AIC    LRT Pr(>Chi)
## <none>                 85.026 113.03
## MDVP.Fhi.Hz.       1   86.356 112.36 1.3304  0.24873
## MDVP.Flo.Hz.       1   86.056 112.06 1.0309  0.30995
## MDVP.Fo.Hz.:RPDE   1   85.204 111.20 0.1789  0.67235
## MDVP.Fo.Hz.:D2     1   88.974 114.97 3.9482  0.04692 *
## RPDE:D2            1   88.460 114.46 3.4341  0.06386 .
## DFA:spread2        1   86.452 112.45 1.4262  0.23239
```

```
## DFA:D2            1   85.942 111.94 0.9161  0.33851
## spread2:D2        1   87.543 113.54 2.5171  0.11262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We continue with this process as we are now seeing some significant interaction effects. We remove `MDVP.Fo.Hz.:RPDE`.

```
inter.model <- update(inter.model, .~. - MDVP.Fo.Hz.:RPDE)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:D2 + RPDE:D2 + DFA:spread2 + DFA:D2 +
##     spread2:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              85.204 111.20
## MDVP.Fhi.Hz.    1   86.582 110.58 1.3774  0.24054
## MDVP.Flo.Hz.    1   86.604 110.60 1.4001  0.23671
## MDVP.Fo.Hz.:D2  1   89.046 113.05 3.8414  0.05000 .
## RPDE:D2         1   88.460 112.46 3.2556  0.07118 .
## DFA:spread2     1   86.799 110.80 1.5951  0.20660
## DFA:D2          1   86.122 110.12 0.9171  0.33823
## spread2:D2      1   87.745 111.75 2.5404  0.11097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we move remove the interaction effect for `DFA:D2`.

```
inter.model <- update(inter.model, .~. - DFA:D2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2 + MDVP.Fo.Hz.:D2 + RPDE:D2 + DFA:spread2 + spread2:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              86.122 110.12
## MDVP.Fhi.Hz.    1   87.521 109.52 1.3993  0.23684
## MDVP.Flo.Hz.    1   87.696 109.70 1.5742  0.20959
## MDVP.Fo.Hz.:D2  1   89.079 111.08 2.9578  0.08547 .
## RPDE:D2         1   88.464 110.46 2.3429  0.12585
## DFA:spread2     1   87.885 109.89 1.7633  0.18421
## spread2:D2      1   87.745 109.75 1.6238  0.20257
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we move remove the main effect for `MDVP.Fhi.Hz.`. We continue to also check the summary of the model to see if the standard errors have inflated. However, our estimates and standard errors are still relatively low.

```
inter.model <- update(inter.model, .~. - MDVP.Fhi.Hz.)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
```

```
##
## Model:
## status ~ MDVP.Fo.Hz. + MDVP.Flo.Hz. + RPDE + DFA + spread2 +
##     D2 + MDVP.Fo.Hz.:D2 + RPDE:D2 + DFA:spread2 + spread2:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              87.521 109.52
## MDVP.Flo.Hz.    1   88.896 108.90 1.3754  0.24088
## MDVP.Fo.Hz.:D2  1   90.206 110.21 2.6854  0.10127
## RPDE:D2         1   89.657 109.66 2.1365  0.14383
## DFA:spread2     1   90.343 110.34 2.8222  0.09297 .
## spread2:D2      1   89.006 109.01 1.4853  0.22295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
inter.model <- update(inter.model, .~. - MDVP.Flo.Hz.)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2 + MDVP.Fo.Hz.:D2 +
##     RPDE:D2 + DFA:spread2 + spread2:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              88.896 108.90
## MDVP.Fo.Hz.:D2  1   90.844 108.84 1.9478  0.16283
## RPDE:D2         1   90.639 108.64 1.7425  0.18683
## DFA:spread2     1   91.802 109.80 2.9058  0.08826 .
## spread2:D2      1   90.098 108.10 1.2013  0.27307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
inter.model <- update(inter.model, .~. - spread2:D2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2 + MDVP.Fo.Hz.:D2 +
##     RPDE:D2 + DFA:spread2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              90.098 108.10
## MDVP.Fo.Hz.:D2  1   92.279 108.28 2.1819  0.13964
## RPDE:D2         1   91.443 107.44 1.3455  0.24607
## DFA:spread2     1   96.257 112.26 6.1592  0.01307 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
inter.model <- update(inter.model, .~. - DFA:spread2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2 + MDVP.Fo.Hz.:D2 +
##     RPDE:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
```

```
## <none>              96.257 112.26
## DFA             1  101.693 115.69 5.4366 0.019719 *
## spread2         1  103.639 117.64 7.3823 0.006587 **
## MDVP.Fo.Hz.:D2  1   99.445 113.44 3.1881 0.074176 .
## RPDE:D2         1   97.485 111.48 1.2280 0.267796
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
inter.model <- update(inter.model, .~. - RPDE:D2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + RPDE + DFA + spread2 + D2 + MDVP.Fo.Hz.:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              97.485 111.48
## RPDE            1   98.599 110.60 1.1147  0.29107
## DFA             1  103.771 115.77 6.2864  0.01217 *
## spread2         1  103.645 115.64 6.1603  0.01306 *
## MDVP.Fo.Hz.:D2  1   99.595 111.59 2.1103  0.14631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
inter.model <- update(inter.model, .~. - RPDE)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + DFA + spread2 + D2 + MDVP.Fo.Hz.:D2
##                Df Deviance    AIC    LRT Pr(>Chi)
## <none>              98.599 110.60
## DFA             1  103.785 113.78 5.1852 0.022780 *
## spread2         1  106.021 116.02 7.4218 0.006444 **
## MDVP.Fo.Hz.:D2  1  100.782 110.78 2.1828 0.139556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The last interaction term we drop is `MDVP.Fo.Hz.:D2`. It appears that none of our interaction terms were significant.

```
inter.model <- update(inter.model, .~. - MDVP.Fo.Hz.:D2)
drop1(inter.model, test='LRT')
```

```
## Single term deletions
##
## Model:
## status ~ MDVP.Fo.Hz. + DFA + spread2 + D2
##             Df Deviance    AIC     LRT  Pr(>Chi)
## <none>           100.78 110.78
## MDVP.Fo.Hz.  1   109.88 117.88  9.1006 0.0025553 **
## DFA          1   105.45 113.45  4.6681 0.0307273 *
## spread2      1   107.75 115.75  6.9626 0.0083229 **
## D2           1   113.53 121.53 12.7457 0.0003568 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(inter.model)
```

```
##
## Call:
## glm(formula = status ~ MDVP.Fo.Hz. + DFA + spread2 + D2, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8850   0.0473   0.2250   0.5424   1.5019
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.1295     0.3670   5.802 6.54e-09 ***
## MDVP.Fo.Hz.  -0.8622     0.3066  -2.812  0.00492 **
## DFA           0.6806     0.3243   2.099  0.03586 *
## spread2       0.9564     0.3842   2.489  0.01281 *
## D2            1.2642     0.4072   3.104  0.00191 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 172.52  on 158  degrees of freedom
## Residual deviance: 100.78  on 154  degrees of freedom
## AIC: 110.78
##
## Number of Fisher Scoring iterations: 6
```

We can see that we do not have high standard errors and beta estimates. We also have all significant main effects left in our model so our final model is `inter.model`.

Full Model Analysis

Firstly, we take a look at our full model to see which variables are significant.

```
full.model <- glm(status  ~ ., data=train, family=binomial)
summary(full.model)
```

```
##
## Call:
## glm(formula = status ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.93601   0.03994   0.20401   0.50144   1.48627
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.2052     0.3928   5.614 1.98e-08 ***
## MDVP.Fo.Hz.   -0.4866     0.3935  -1.237  0.21623
## MDVP.Fhi.Hz.  -0.3550     0.2446  -1.451  0.14672
## MDVP.Flo.Hz.  -0.3650     0.3404  -1.072  0.28359
## RPDE           0.2582     0.3370   0.766  0.44360
## DFA            0.7497     0.3621   2.071  0.03839 *
```

```
## spread2        0.9175     0.3968   2.312  0.02076 *
## D2             1.3391     0.4516   2.965  0.00303 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 172.522  on 158  degrees of freedom
## Residual deviance:  96.632  on 151  degrees of freedom
## AIC: 112.63
##
## Number of Fisher Scoring iterations: 6
```

We also observe that we have that `DFA`, `spread2` and `D2` and our intercept are our variables that are statistically significant. We retrain `model1` based on these significant features.

```
model1 <- glm(status  ~ DFA + spread2 + D2, data=train, family=binomial)
```

Stepwise AIC

We use stepwise AIC with backwards selection to create an alternative model and see if our final models are similar. We also allow for interaction terms using the `scope` parameter.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
step.model <- full.model %>% stepAIC(trace=TRUE, scope = . ~ .^2, direction='backward')
```

```
## Start:  AIC=112.63
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2
##
##                Df Deviance    AIC
## - RPDE          1   97.226 111.23
## - MDVP.Flo.Hz.  1   97.821 111.82
## - MDVP.Fo.Hz.   1   98.173 112.17
## <none>              96.632 112.63
## - MDVP.Fhi.Hz.  1   98.658 112.66
## - DFA           1  101.245 115.25
## - spread2       1  102.535 116.53
## - D2            1  108.315 122.31
##
## Step:  AIC=111.23
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + DFA + spread2 +
##     D2
##
##                Df Deviance    AIC
## - MDVP.Flo.Hz.  1   99.179 111.18
## - MDVP.Fhi.Hz.  1   99.208 111.21
## <none>              97.226 111.23
## - MDVP.Fo.Hz.   1   99.227 111.23
```

```
## - DFA           1  101.260 113.26
## - spread2       1  104.288 116.29
## - D2            1  108.332 120.33
##
## Step:  AIC=111.18
## status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + DFA + spread2 + D2
##
##                Df Deviance    AIC
## - MDVP.Fhi.Hz.  1  100.782 110.78
## <none>              99.179 111.18
## - DFA           1  102.258 112.26
## - spread2       1  106.185 116.19
## - MDVP.Fo.Hz.   1  107.028 117.03
## - D2            1  112.658 122.66
##
## Step:  AIC=110.78
## status ~ MDVP.Fo.Hz. + DFA + spread2 + D2
##
##               Df Deviance    AIC
## <none>             100.78 110.78
## - DFA          1   105.45 113.45
## - spread2      1   107.75 115.75
## - MDVP.Fo.Hz.  1   109.88 117.88
## - D2           1   113.53 121.53
```

From here, we have that `MDVP.Fo.Hz.` + `DFA` + `spread2` + `D2` are significant variables. We can test which of these two models is preferred.

```
model2 <- glm(status  ~ MDVP.Fo.Hz. + DFA + spread2 + D2, data=train, family=binomial)
```

Analysis

Model Comparison

First, we compare `model1` and `model2` against the `full.model` using an ANOVA test since these models are nested. This will help us to decide which model fits the data better.

```
anova(model1, full.model)
```

```
## Analysis of Deviance Table
##
## Model 1: status ~ DFA + spread2 + D2
## Model 2: status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2
##   Resid. Df Resid. Dev Df Deviance
## 1       155    109.883
## 2       151     96.632  4   13.251
```

Then, we run a chi-squared test to determine which is a better fit.

```
pchisq(deviance(model1) - deviance(full.model), df.residual(model1) - df.residual(full.model))
```

```
## [1] 0.9898866
```

We have a high p-value meaning that we do not have evidence to reject our null hypothesis that `model1` is a better fit than `full.model`. Thus, since `model1` is simpler, we prefer this over our `full.model`.

```
anova(model2, full.model)
```

```
## Analysis of Deviance Table
##
## Model 1: status ~ MDVP.Fo.Hz. + DFA + spread2 + D2
## Model 2: status ~ MDVP.Fo.Hz. + MDVP.Fhi.Hz. + MDVP.Flo.Hz. + RPDE + DFA +
##     spread2 + D2
##   Resid. Df Resid. Dev Df Deviance
## 1       154    100.782
## 2       151     96.632  3   4.1502
```

```
pchisq(deviance(model2) - deviance(full.model), df.residual(model2) - df.residual(full.model))
```

```
## [1] 0.7543024
```

Again, we have a high p-value meaning that we do not have evidence to reject our null hypothesis that `model2` is a better fit than `full.model`. Thus, since `model2` is simpler, we prefer this over our `full.model`.

Next, we compare `model1` vs. `model2`.

```
anova(model1, model2)
```

```
## Analysis of Deviance Table
##
## Model 1: status ~ DFA + spread2 + D2
## Model 2: status ~ MDVP.Fo.Hz. + DFA + spread2 + D2
##   Resid. Df Resid. Dev Df Deviance
## 1       155     109.88
## 2       154     100.78  1   9.1006
```

```
pchisq(deviance(model1) - deviance(model2), df.residual(model1) - df.residual(model2))
```

```
## [1] 0.9974447
```

Again, we have a high p-value so the two models are equivalently good at capturing meaningful information from the data. Now, since our stepAIC and LRT methods both resulted in `model2`, we decide to use this as our final model.

Diagnostic Plots

Now, we want to check the logistic regression assumptions with our final model. This is used to verify that the logistic regression model is a good fit to our data in addition to using step AIC. These include: 1. Linearity assumption 2. Lack of strongly influential outliers 3. Absence of Multicollinearity

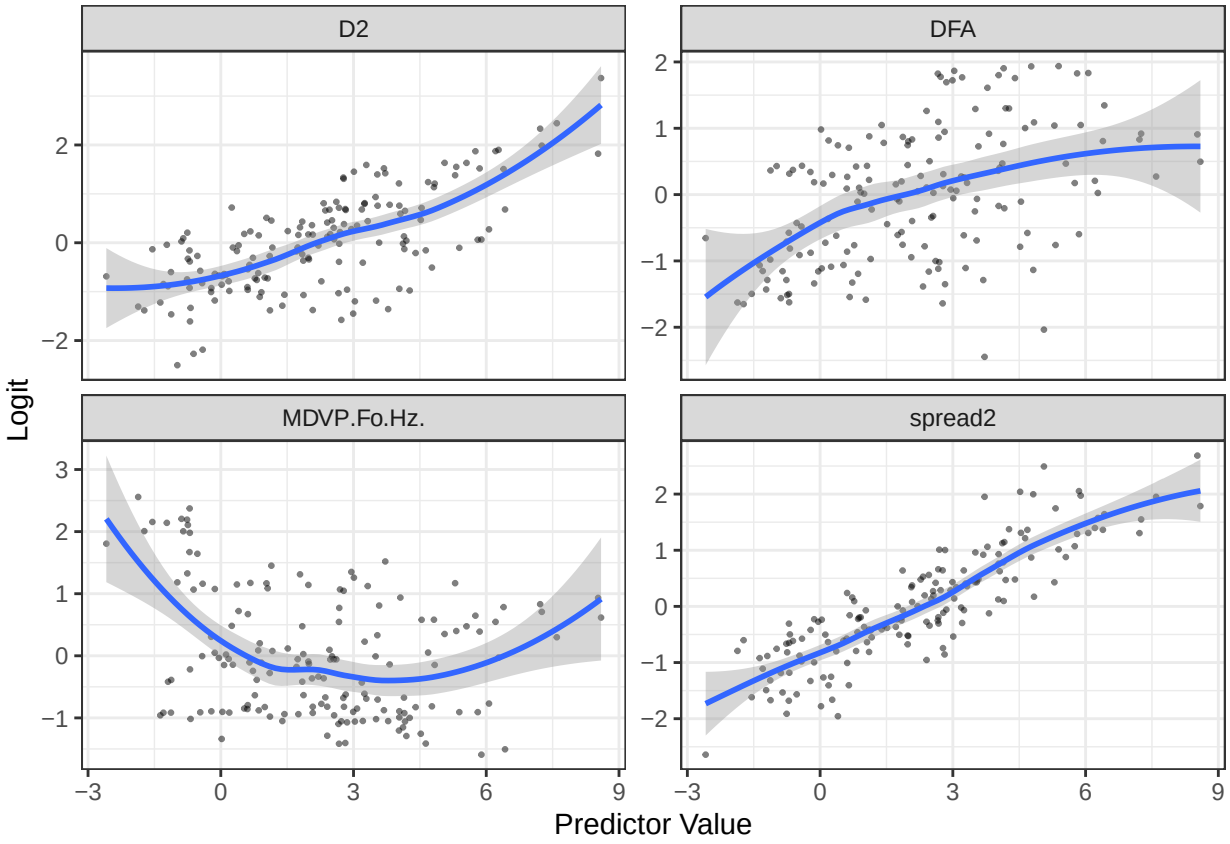
Linearity Assumption

```
# Get our logit values
predictors <- c('MDVP.Fo.Hz.', 'DFA', 'spread2', 'D2')

probabilities <- predict(model1, type = "response")

model.x <- train.data %>%
  dplyr::select(all_of(predictors)) %>%
  mutate(logit=log(probabilities/(1-probabilities))) %>%
  gather(key="predictors", value="predictor.value", -logit)
```

```
ggplot(model.x, aes(logit, predictor.value))+
  geom_point(size = 0.5, alpha = 0.5) +
  geom_smooth(method = "loess") +
```

```
  theme_bw() +
  facet_wrap(~predictors, scales = "free_y") +
  labs(y="Logit", x="Predictor Value")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

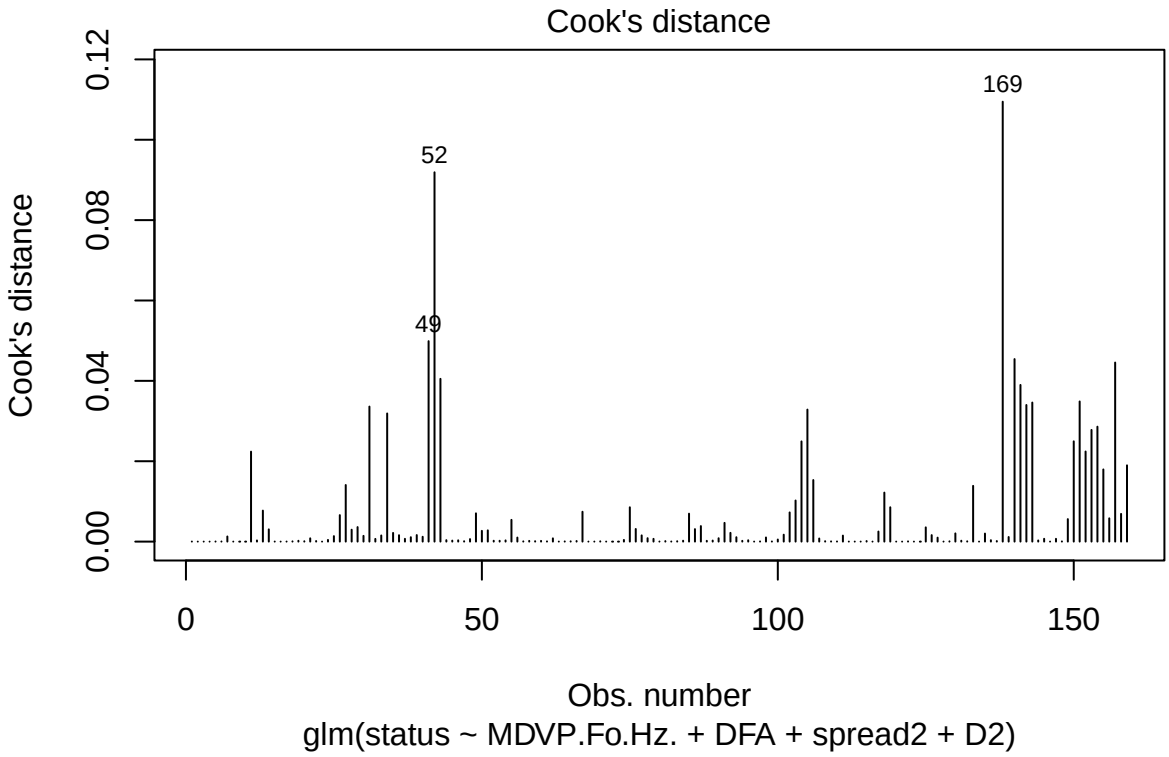


We see that the smoothing curve of MDVP.Fo.Hz., D2, DFA, and spread2 are relatively linear. `MDVP.Fo.Hz.` may be better transformed by a polynomial but for our purposes, we do not add further polynomial terms into our model to avoid multicollinearity. Next, we check for outliers.

Influential Values

We check for these values since they can alter the quality of our model using Cook's distance plot.

```
plot(model2, which = 4, id.n = 3)
```



To check whether the indices 49, 52 and 169 are influential observations, we need to check their standardized residual error.

```
library(broom)
```

```
model2.data <- augment(model2) %>% mutate(index= 1:n())
```

The top 3 values based on Cook's distance are:

```
model2.data %>% top_n(3, .cooksd)
```

```
## # A tibble: 3 x 13
##   .rownames status MDVP.Fo~1   DFA spread2     D2 .fitted .resid .std.~2    .hat
##   <chr>     <fct>      <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>   <dbl>
## 1 49        0         -0.774 0.281   0.478 -0.789    2.45  -2.25   -2.27 0.0207
## 2 52        0         -0.674 0.764   0.785  0.131    4.15  -2.88   -2.90 0.00717
## 3 169       0          1.05  0.129  -0.743  1.31     2.26  -2.17   -2.23 0.0516
## # ... with 3 more variables: .sigma <dbl>, .cooksd <dbl>, index <int>, and
## #   abbreviated variable names 1: MDVP.Fo.Hz., 2: .std.resid
```

This gives us our information on the outliers that we saw in the Cook's distance plot. Then, we check if any of the standardized residuals are greater than 3.

```
model2.data %>% filter(abs(.std.resid) > 3)
```

```
## # A tibble: 0 x 13
## # ... with 13 variables: .rownames <chr>, status <fct>, MDVP.Fo.Hz. <dbl>,
## #   DFA <dbl>, spread2 <dbl>, D2 <dbl>, .fitted <dbl>, .resid <dbl>,
## #   .std.resid <dbl>, .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, index <int>
```

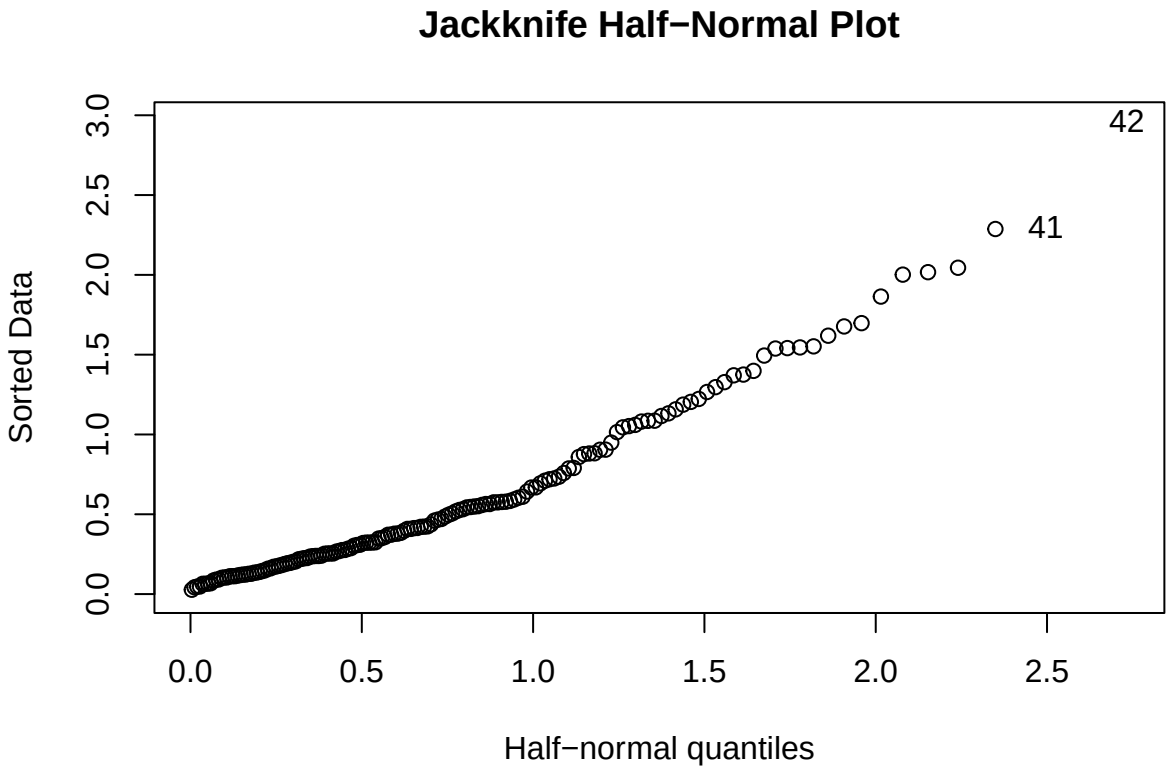
None of our variables have standard residuals outside of the (-3, 3) range; hence, we have no influential observations in our data. Another plot to check for outlieres is to use the jackman knife plot.

```
library(faraway)
```

```
##
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
halfnorm(rstudent(model2), main='Jackknife Half-Normal Plot')
```



We check if 41 and 42 have high standardized residuals.

```
model2.data[model2.data$.rownames == 41,]$.std.resid
```

```
## [1] 0.5826458
```

```
model2.data[model2.data$.rownames == 42,]$.std.resid
```

```
## [1] 1.227881
```

Again, we do not have large standardized residuals so these are not outliers. Lastly, we dealt with multicollinearity earlier by removing variables that were highly correlated with one another. We will do another check to ensure our logistic regression model holds.

Multicollinearity

```
car::vif(model2)
```

```
## MDVP.Fo.Hz.         DFA     spread2          D2
##    1.290146    1.165339    1.130790    1.299703
```

We are checking the variance inflation factor (VIF) which measures the strength of correlation between independent variables in a regression analysis. In this case, our VIF values are relatively low (close to 1) so

we do not have a multicollinearity issue.

Lastly, we take a look at the exponentiation of our coefficients to understand the statistical significance and effect they have on distinguishing PD.

```
exp(summary(model2)$coefficients[, 'Estimate'])
```

```
## (Intercept) MDVP.Fo.Hz.         DFA     spread2          D2
##   8.4103653   0.4222183   1.9750146   2.6023843   3.5401906
```

Hence, we see that for every scenario, a one unit increase in the speech signal increases the odds ratio of having PD. We can also get the 95% confidence interval of these estimates.

```
exp(confint(model2))
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) 4.4124676 18.929642
## MDVP.Fo.Hz. 0.2213961  0.747110
## DFA         1.0641821  3.844342
## spread2     1.2680000  5.794595
## D2          1.7037134  8.567507
```

Model Assessment

Now that we have validated that the logistic regression model is adequate, we report a few metrics to assess the performance of our model. We begin by predicting on our testing data.

```
pred <- predict(model2, newdata=test.data, type='response')
pred_class <- as.integer(pred >= 0.5)
```

Then, since we are building a classification model, we take a look at our confusion matrix. We also want to calculate precision, recall, and f1-score for our model.

```
library(caret)
```

```
confusionMatrix(data=factor(pred_class), reference=factor(test.y), mode='prec_recall', positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  4  1
##          1  7 24
##
##                Accuracy : 0.7778
##                  95% CI : (0.6085, 0.8988)
##     No Information Rate : 0.6944
##     P-Value [Acc > NIR] : 0.1841
##
##                   Kappa : 0.382
##
##  Mcnemar's Test P-Value : 0.0771
##
##               Precision : 0.7742
##                  Recall : 0.9600
##                      F1 : 0.8571
##              Prevalence : 0.6944
```

```
##          Detection Rate : 0.6667
##    Detection Prevalence : 0.8611
##       Balanced Accuracy : 0.6618
##
##        'Positive' Class : 1
##
```

Looking at our confusion matrix, we do have a decent number of correct predictions. However, we note that we do predict 1 when the expected value is 0 a total of 7 times which is larger than the number of correct predictions for 0. This is likely due to our imbalanced data as our model is more familiar with the data corresponding to an expected value of 1.

We can also see that we have a good recall score, and a decent precision and f1-score. This aligns with what our confusion matrix showed us that our classifier is good at predicting Parkinson's cases but not those without Parkinson's.

Future extensions could include training a SVM model or alternative classifiers to improve the accuracy of this predictor.