Lab 7

Isaac Plotkin

3/5/2022

1.

```
spotify_data <- read_csv("spotify.csv") %>%
  drop_na() %>% #remove observations with missing values
  mutate(key = case_when(
    key == 2 ~ "D",
    key == 3 ~ "D#",
    TRUE ~ "Other"
  ),
  target = as.factor(target),
  )
```

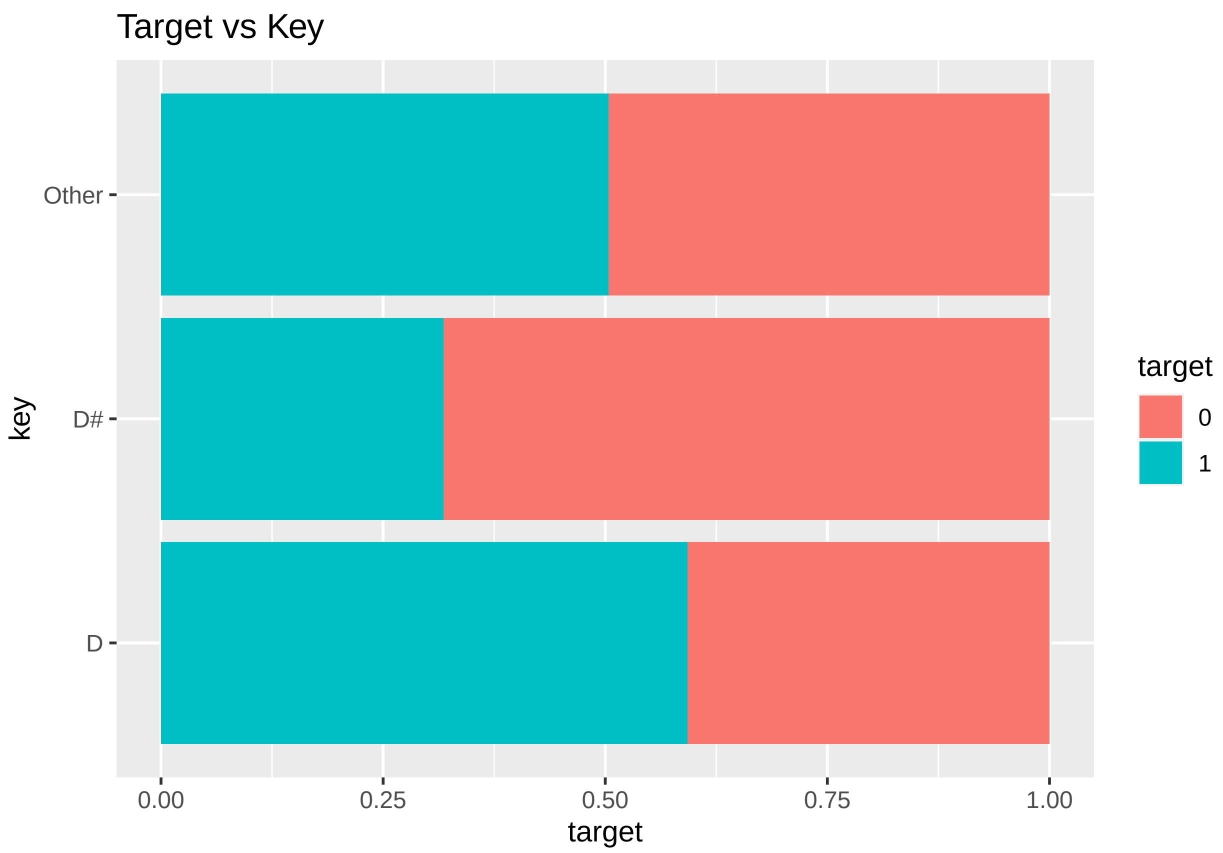
```
## New names:
## * `` -> ...1

## Rows: 2017 Columns: 17

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): song_title, artist
## dbl (15): ...1, acousticness, danceability, duration_ms, energy, instrumenta...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ggplot(data = spotify_data, aes(x = key, fill = target)) +
  geom_bar(position = "fill") +
  labs(y = "target", title = "Target vs Key") +
  coord_flip()
```



2.

```
target_model <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness + loudness +
                speechiness + valence, data = spotify_data, family = binomial)

tidy(target_model, conf.int = TRUE, exponentiate = FALSE)
```

```
## # A tibble: 8 x 7
##   term                estimate   std.error statistic  p.value conf.low conf.high
##   <chr>                  <dbl>       <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)      -2.96       0.276          -10.7  1.10e-26 -3.50e+0  -2.42e+0
## 2 acousticness     -1.72       0.240           -7.18 6.89e-13 -2.20e+0  -1.26e+0
## 3 danceability      1.63       0.344            4.74 2.17e- 6  9.58e-1   2.31e+0
## 4 duration_ms       0.00000287 0.000000680      4.23 2.39e- 5  1.56e-6   4.23e-6
## 5 instrumentalness  1.35       0.207            6.55 5.80e-11  9.52e-1   1.76e+0
## 6 loudness         -0.0874     0.0173          -5.06 4.14e- 7 -1.22e-1  -5.38e-2
## 7 speechiness       4.07       0.583            6.98 2.85e-12  2.95e+0   5.23e+0
## 8 valence           0.856      0.223            3.84 1.25e- 4  4.20e-1   1.30e+0
```

3.

```
target_key_model <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness +
                        loudness + speechiness + valence + key,
                        data = spotify_data, family = binomial)
```

```
tidy(target_key_model, conf.int = TRUE, exponentiate = FALSE)
```

```
## # A tibble: 10 x 7
##    term                estimate  std.error statistic  p.value conf.low conf.high
##    <chr>                  <dbl>      <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 (Intercept)      -2.51          3.11e-1     -8.07 7.14e-16 -3.12e+0  -1.90e+0
##  2 acousticness     -1.70          2.41e-1     -7.07 1.60e-12 -2.18e+0  -1.23e+0
##  3 danceability      1.65          3.45e-1      4.77 1.80e- 6  9.75e-1   2.33e+0
##  4 duration_ms       0.00000286    6.84e-7      4.19 2.82e- 5  1.55e-6   4.23e-6
##  5 instrumentalness  1.38          2.07e-1      6.67 2.60e-11  9.81e-1   1.80e+0
##  6 loudness         -0.0866        1.73e-2     -5.02 5.21e- 7 -1.21e-1  -5.30e-2
##  7 speechiness       4.03          5.85e-1      6.90 5.33e-12  2.90e+0   5.20e+0
##  8 valence           0.881         2.24e-1      3.93 8.61e- 5  4.42e-1   1.32e+0
##  9 keyD#            -1.07          3.35e-1     -3.20 1.36e- 3 -1.75e+0  -4.28e-1
## 10 keyOther         -0.494         1.69e-1     -2.92 3.47e- 3 -8.28e-1  -1.65e-1
```

```
anova(target_model, target_key_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ acousticness + danceability + duration_ms + instrumentalness +
##     loudness + speechiness + valence
## Model 2: target ~ acousticness + danceability + duration_ms + instrumentalness +
##     loudness + speechiness + valence + key
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2009     2518.5
## 2      2007     2505.2  2   13.357 0.001258 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

4.

```
tidy(target_key_model, conf.int = TRUE, exponentiate = FALSE)
```

```
## # A tibble: 10 x 7
##    term                estimate  std.error statistic  p.value conf.low conf.high
##    <chr>                  <dbl>      <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 (Intercept)      -2.51          3.11e-1     -8.07 7.14e-16 -3.12e+0  -1.90e+0
##  2 acousticness     -1.70          2.41e-1     -7.07 1.60e-12 -2.18e+0  -1.23e+0
##  3 danceability      1.65          3.45e-1      4.77 1.80e- 6  9.75e-1   2.33e+0
##  4 duration_ms       0.00000286    6.84e-7      4.19 2.82e- 5  1.55e-6   4.23e-6
##  5 instrumentalness  1.38          2.07e-1      6.67 2.60e-11  9.81e-1   1.80e+0
##  6 loudness         -0.0866        1.73e-2     -5.02 5.21e- 7 -1.21e-1  -5.30e-2
##  7 speechiness       4.03          5.85e-1      6.90 5.33e-12  2.90e+0   5.20e+0
##  8 valence           0.881         2.24e-1      3.93 8.61e- 5  4.42e-1   1.32e+0
##  9 keyD#            -1.07          3.35e-1     -3.20 1.36e- 3 -1.75e+0  -4.28e-1
## 10 keyOther         -0.494         1.69e-1     -2.92 3.47e- 3 -8.28e-1  -1.65e-1
```

keyD# shows that the target score decreases by -1.07% for every song that uses that key.

5.

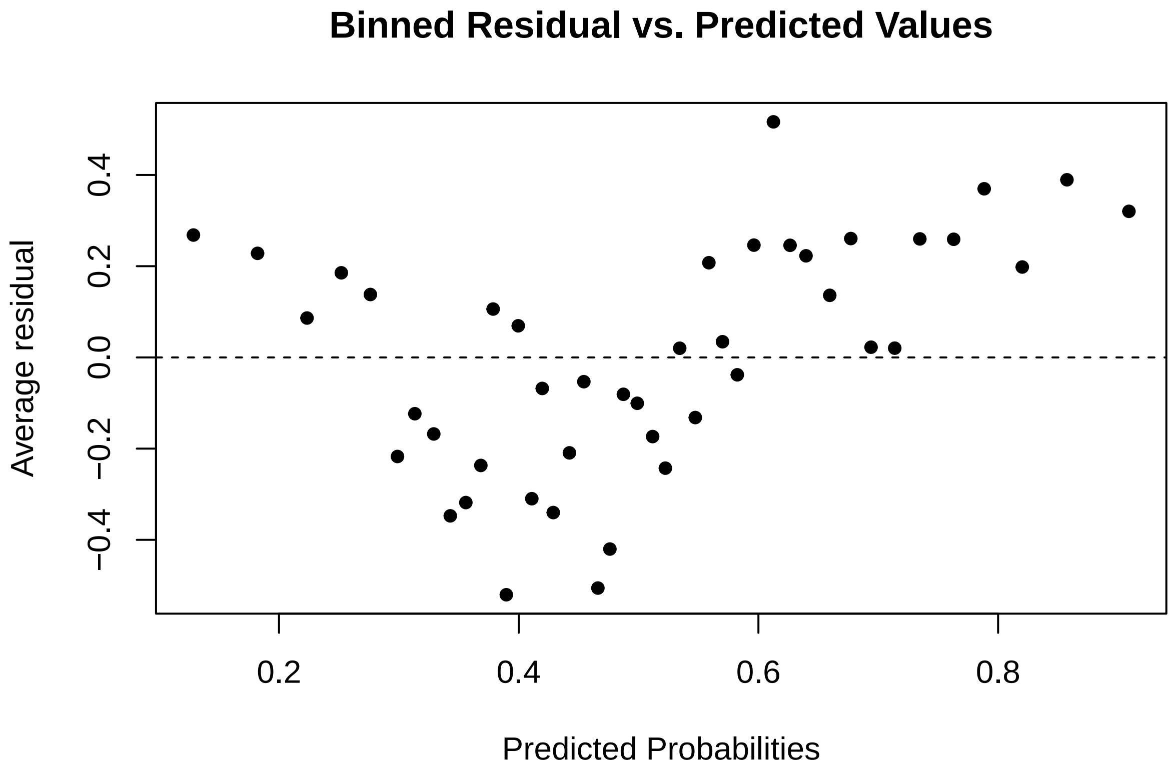
```r
spotify_aug <- augment(target_key_model, type.predict = "response",
                       type.residuals = "deviance")

spotify_aug
```

```
## # A tibble: 2,017 x 15
##    target acousticness danceability duration_ms instrumentalness loudness
##    <fct>         <dbl>        <dbl>       <dbl>            <dbl>    <dbl>
##  1 1           0.0102         0.833      204600       0.0219        -8.80
##  2 1           0.199          0.743      326933       0.00611      -10.4
##  3 1           0.0344         0.838      185707       0.000234      -7.15
##  4 1           0.604          0.494      199413       0.51         -15.2
##  5 1           0.18           0.678      392893       0.512        -11.6
##  6 1           0.00479        0.804      251333       0             -6.68
##  7 1           0.0145         0.739      241400       0.00000727   -11.2
##  8 1           0.0202         0.266      349667       0.664        -11.6
##  9 1           0.0481         0.603      202853       0             -3.63
## 10 1           0.00208        0.836      226840       0             -7.79
## # ... with 2,007 more rows, and 9 more variables: speechiness <dbl>,
## #   valence <dbl>, key <chr>, .fitted <dbl>, .resid <dbl>, .std.resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>
```

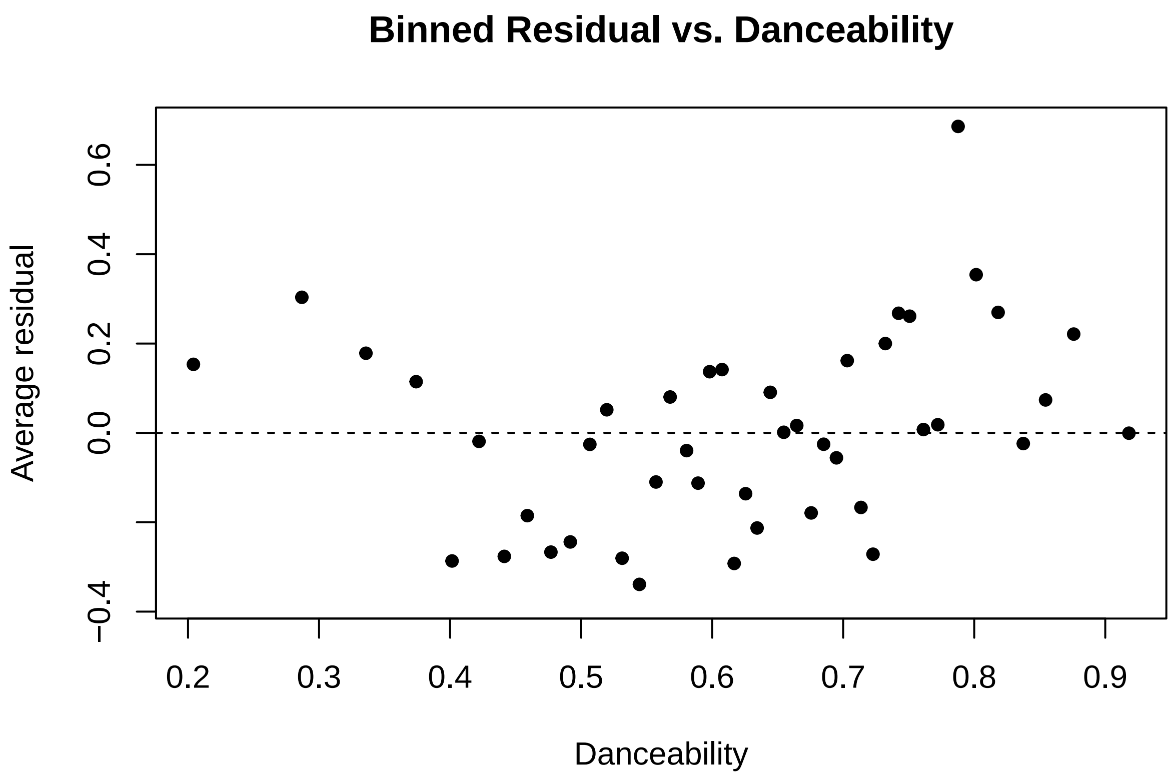
6.

```r
arm::binnedplot(x = spotify_aug$.fitted, y = spotify_aug$.resid,
                xlab = "Predicted Probabilities",
                main = "Binned Residual vs. Predicted Values",
                col.int = FALSE)
```



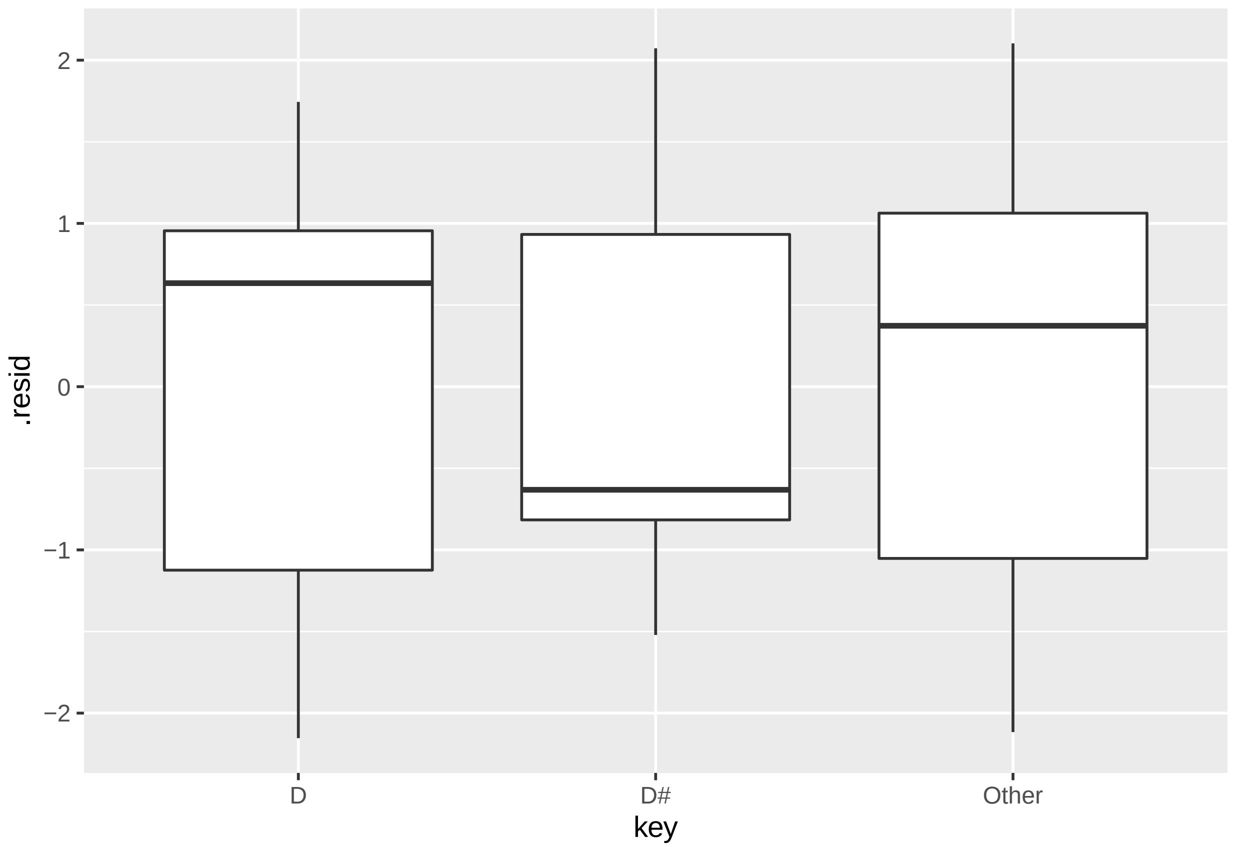
7.

```
arm::binnedplot(x = spotify_aug$danceability, y = spotify_aug$.resid,
                xlab = "Danceability",
                main = "Binned Residual vs. Danceability",
                col.int = FALSE)
```



8.

```
ggplot(data = spotify_aug, aes(x = key, y = .resid)) +
  geom_boxplot()
```

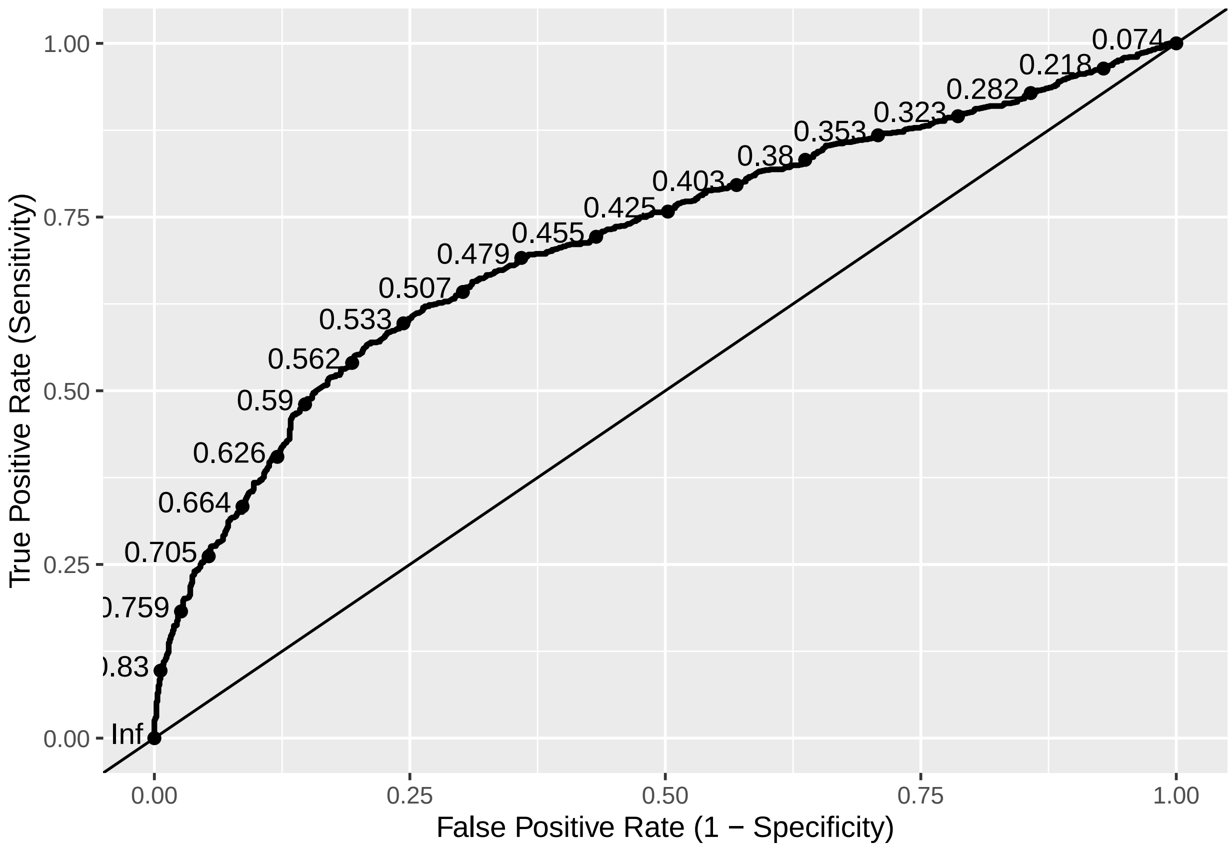


9. The linearity assumption is not satisfied because the binned residual vs predicted values plot does not have a cloud distribution. It has a V shape pattern to it. I also did not test every variable's residual plot for patterns.

Part III: Model Assessment & Prediction

10.

```
(roc_curve <- ggplot(spotify_aug,
                     aes(d = as.numeric(target) - 1,
                         m = .fitted)) +
  geom_roc(n.cuts = 20, labelround = 3) +
  geom_abline(intercept = 0) +
  labs(x = "False Positive Rate (1 - Specificity)",
       y = "True Positive Rate (Sensitivity)") )
```



AUC

```
calc_auc(roc_curve)$AUC
```

```
## [1] 0.7137869
```

11. Yes the model effectively differentiates between the songs the user likes versus those they don't like, but not at a very high accuracy.
12. The best choice for threshold is 0.533 according to the ROC curve.
13.

```
threshold <- 0.533
spotify_aug %>%
  mutate(predict = if_else(.fitted > threshold, "1: Yes", "0: No")) %>%
  group_by(target, predict) %>%
  summarise(n = n()) %>%
  kable(format="markdown")
```

```
## `summarise()` has grouped output by 'target'. You can override using the `.groups` argument.
```

target	predict	n
0	0: No	755

target	predict	n
0	1: Yes	242
1	0: No	412
1	1: Yes	608

14.

- What is the proportion of true positives (sensitivity)? 608 / (608 + 412) = 608 / 1020 = 0.596
- What is the proportion of false positives (1 - specificity)? 242 / (242 + 755) = 242 / 997 = 0.243
- What is the misclassification rate? (242 + 412) / (242 + 412 + 755 + 608) = 654 / 2017 = 0.324