Lab 07

Rohit Kulkarni

3/2/2022

Exercises

Part I: Data Prep & Modeling

1. Read through the Spotify documentation page to learn more about the variables in the dataset. The response variable for this analysis is target, where 1 indicates the user likes the song and 0 otherwise. Let's prepare the response and some predictor variables before modeling.

- If needed, change target so that it is factor variable type in R.

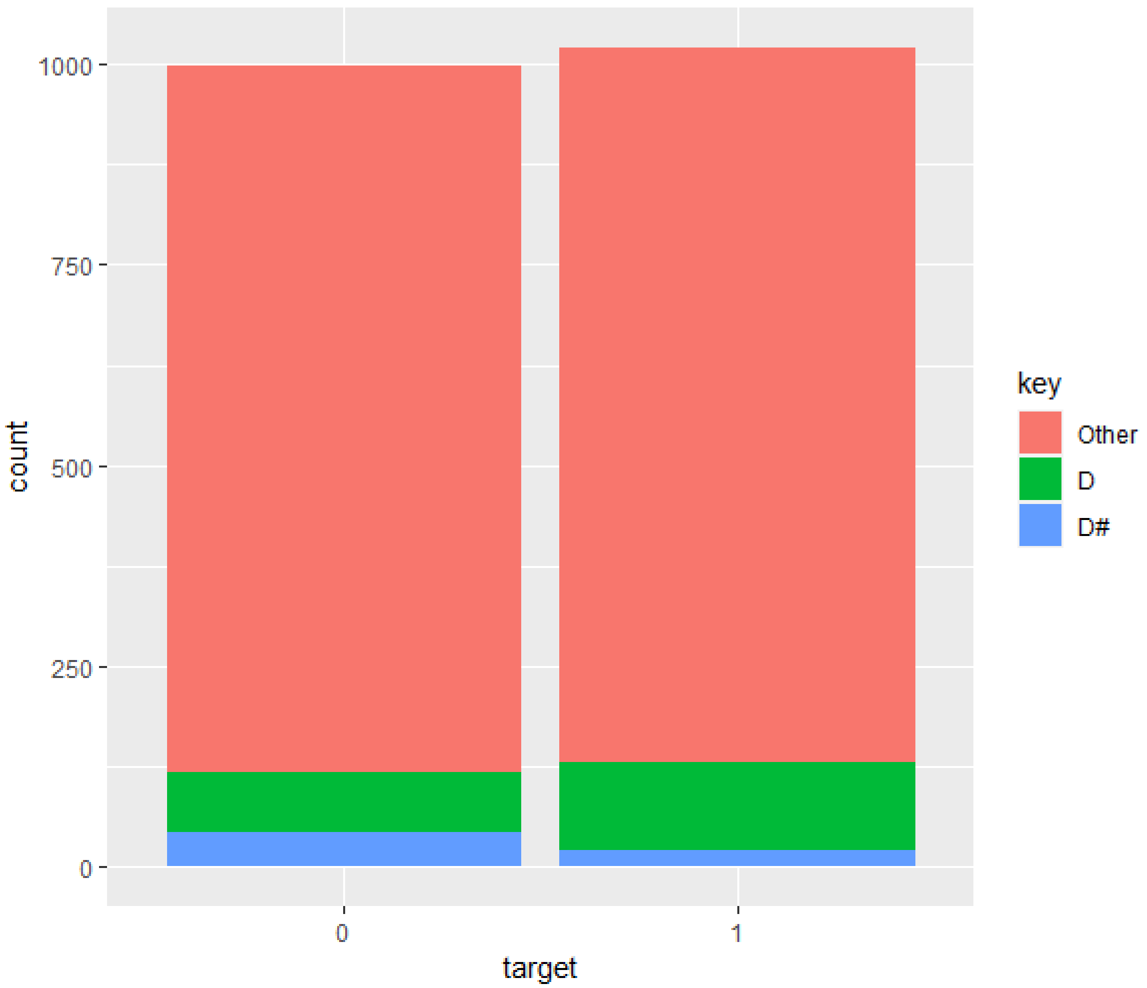
```
spotify$target <- as.factor(spotify$target)
```

- Change key so that it is a factor variable type in R, which takes values "D" if key==2, "D#" if key==3, and "Other" for all other values.

```
spotify$key <- as.factor(spotify$key)
spotify$key <- fct_recode(spotify$key, "D" = "2", "D#" = "3", "Other" = "0", "Other" = "1",          "Ot
```

- Plot the relationship between target and key. Briefly describe the relationship between the two variables.

```
ggplot(spotify, aes(x=target, fill=key)) + geom_bar()
```



Both 1 and 0 factors for target mostly consist of observations with “Other” as the key value. However, in the 1 column, there is a significantly higher proportion of “D” and lower proportion of “D#” than in the 0 column.

2. Fit a logistic regression model with `target` as the response variable and the following as predictors: `acousticness, danceability, duration_ms, instrumentalness, loudness, speechiness, and valence`. Display the model output.

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

Output:

```
Coefficients:
     (Intercept)      acousticness      danceability       duration_ms  instrumentalness
      -2.955e+00        -1.722e+00         1.630e+00         2.871e-06         1.353e+00
        loudness       speechiness           valence
      -8.744e-02         4.072e+00         8.564e-01

Degrees of Freedom: 2016 Total (i.e. Null);  2009 Residual
Null Deviance:      2796
Residual Deviance: 2519     AIC: 2535
```

3. We consider adding `key` to the model. Conduct the appropriate test to determine if `key` should be included in the model. Display the output from the test and write your conclusion in the context of the data.

```
anova(model, model_test, test="Chisq")
```

I performed a Drop-in-Deviance test with these results:

```
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 **
```

The test statistic is 13.357 on 2 degrees of freedom, which produces a p-value of 0.001258. Since the value is less than 0.05, `key` will be included in the model.

Use the model you selected in Exercise 3 for the remainder of the lab.

4. Display the model you chose in Exercise 3. If appropriate, interpret the coefficient for `keyD#` in the context of the data. Otherwise, state why it's not appropriate to interpret this coefficient.

```
summary(model)
```

Output:

```
Coefficients:
     (Intercept)      acousticness      danceability       duration_ms  instrumentalness
      -3.003e+00        -1.702e+00         1.649e+00         2.863e-06         1.383e+00
        loudness       speechiness           valence              keyD             keyD#
      -8.662e-02         4.034e+00         8.809e-01         4.939e-01        -5.793e-01

Degrees of Freedom: 2016 Total (i.e. Null);  2007 Residual
Null Deviance:      2796
Residual Deviance: 2505     AIC: 2525
```

The coefficient for KeyD# is -0.5793 which means that there is a negative correlation between a song having a key of D# and it being a song that the user "likes" (target == 1).

Part II: Checking Assumptions

In the next few questions, we will do an abbreviated analysis of the residuals.

5. Use the `augment` function to calculate the predicted probabilities and corresponding residuals.

```
spotify_aug <- augment(model, spotify)
```

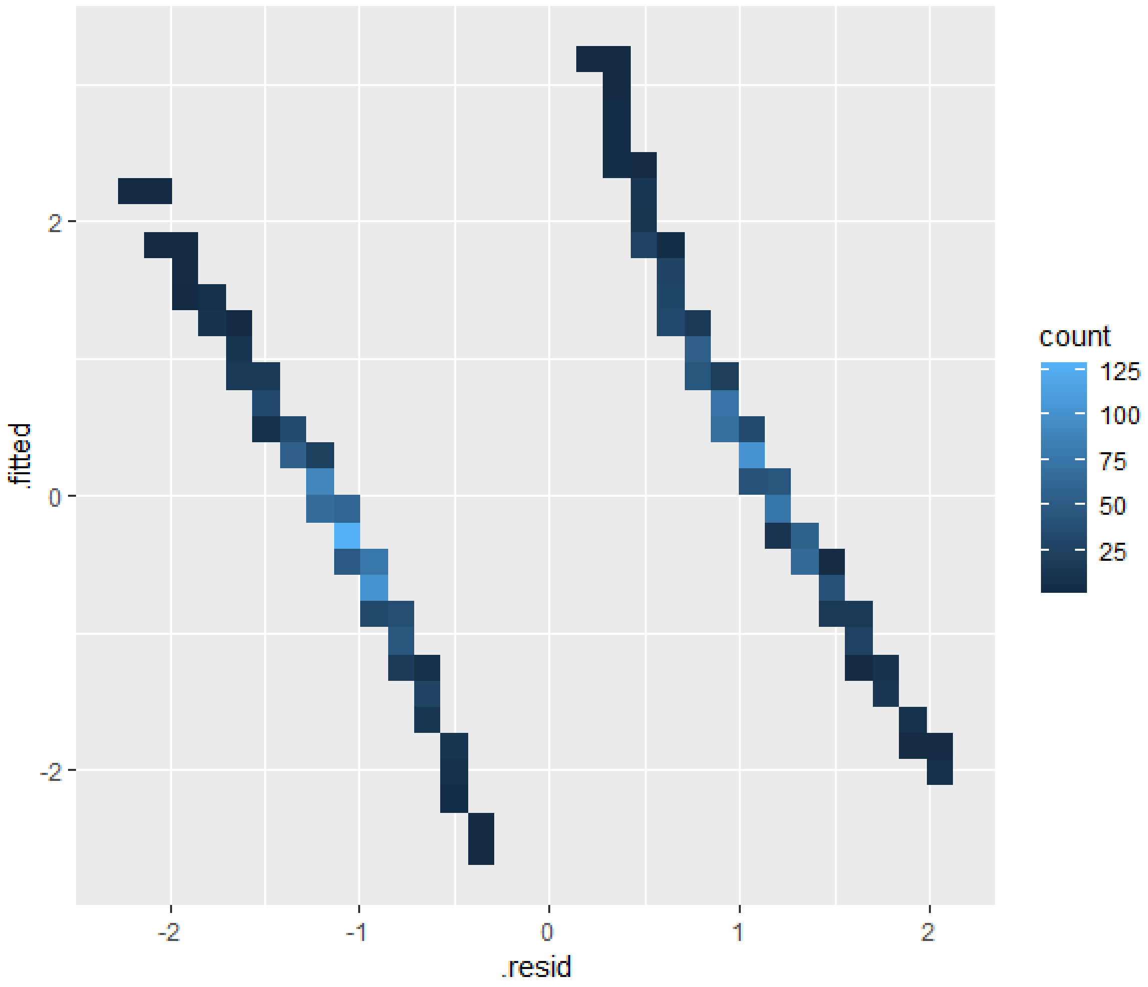


Figure 1: Residuals vs. Predicted Probabilities

6. Create a binned plot of the residuals versus the predicted probabilities.

```
ggplot(spotify_aug, aes(x=.resid, y=.fitted)) + geom_bin2d()
```

7. Choose a quantitative predictor in the final model. Make the appropriate table or plot to examine the residuals versus this predictor variable.

Quantitative predictor: `danceability`

```
ggplot(spotify_aug, aes(x=.resid, y=danceability)) + geom_point()
```

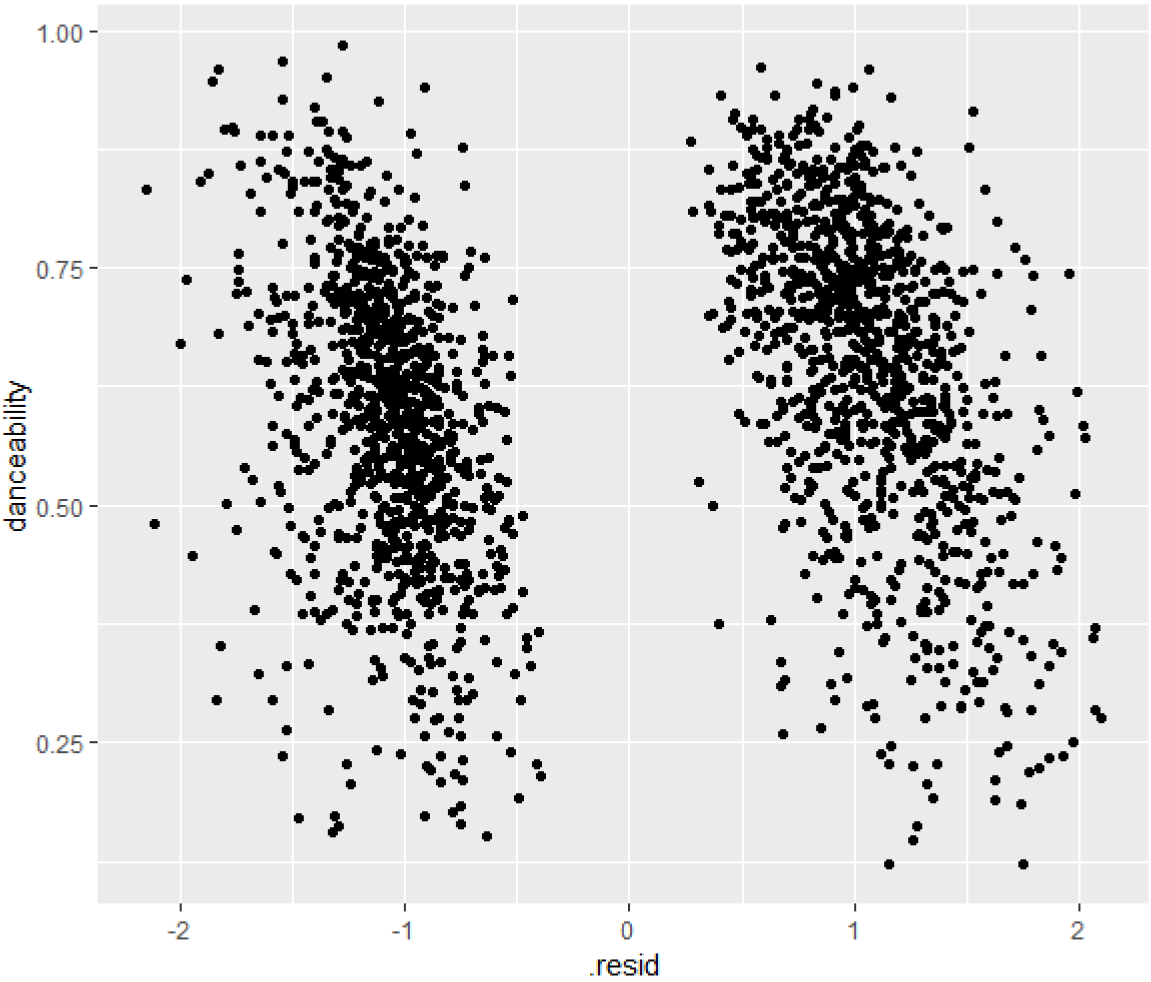


Figure 2: Residuals vs Danceability

8. Choose a categorical predictor in the final model. Make the appropriate table or plot to examine the residuals versus this predictor variable

Categorical Predictor: `key`

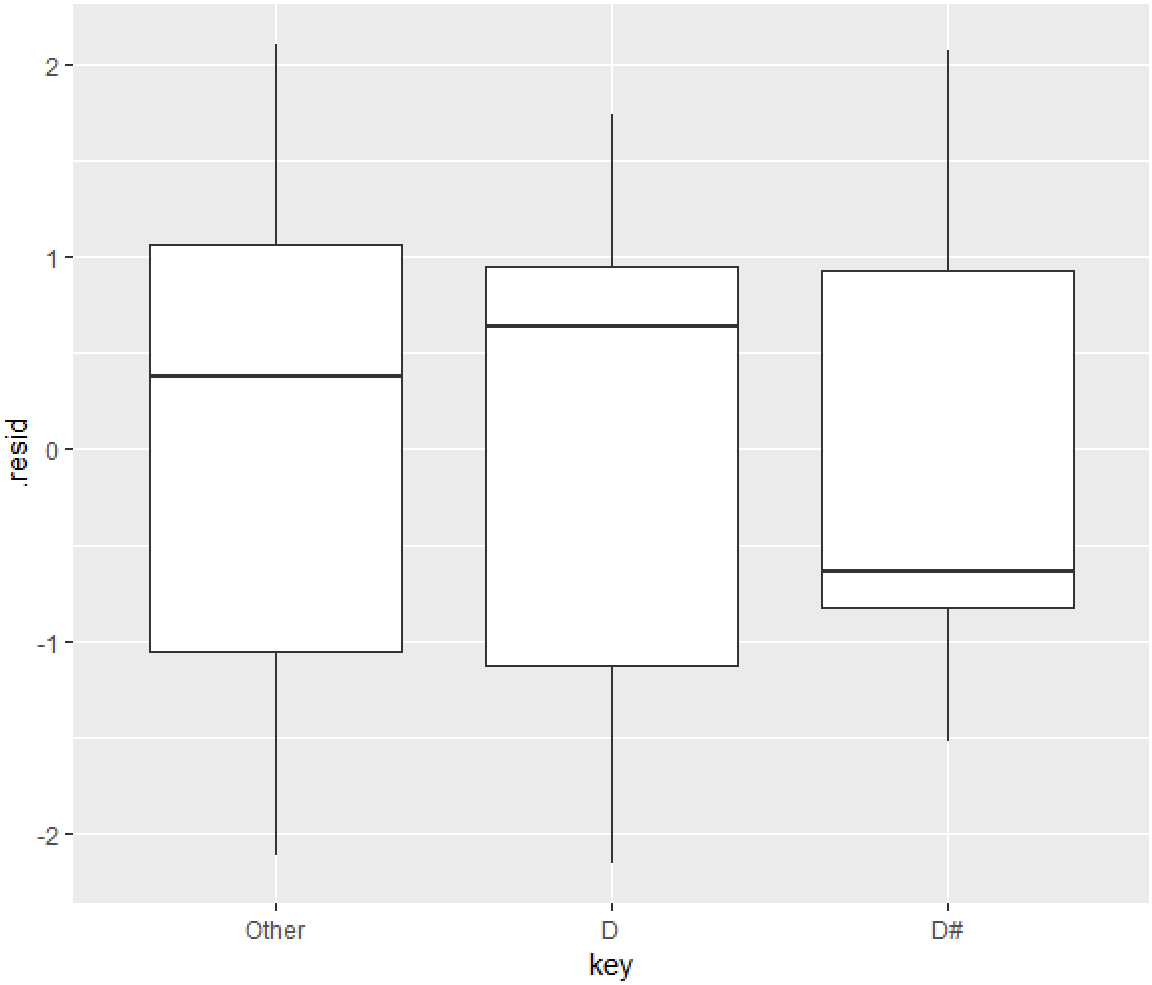


Figure 3: Key vs Residuals

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

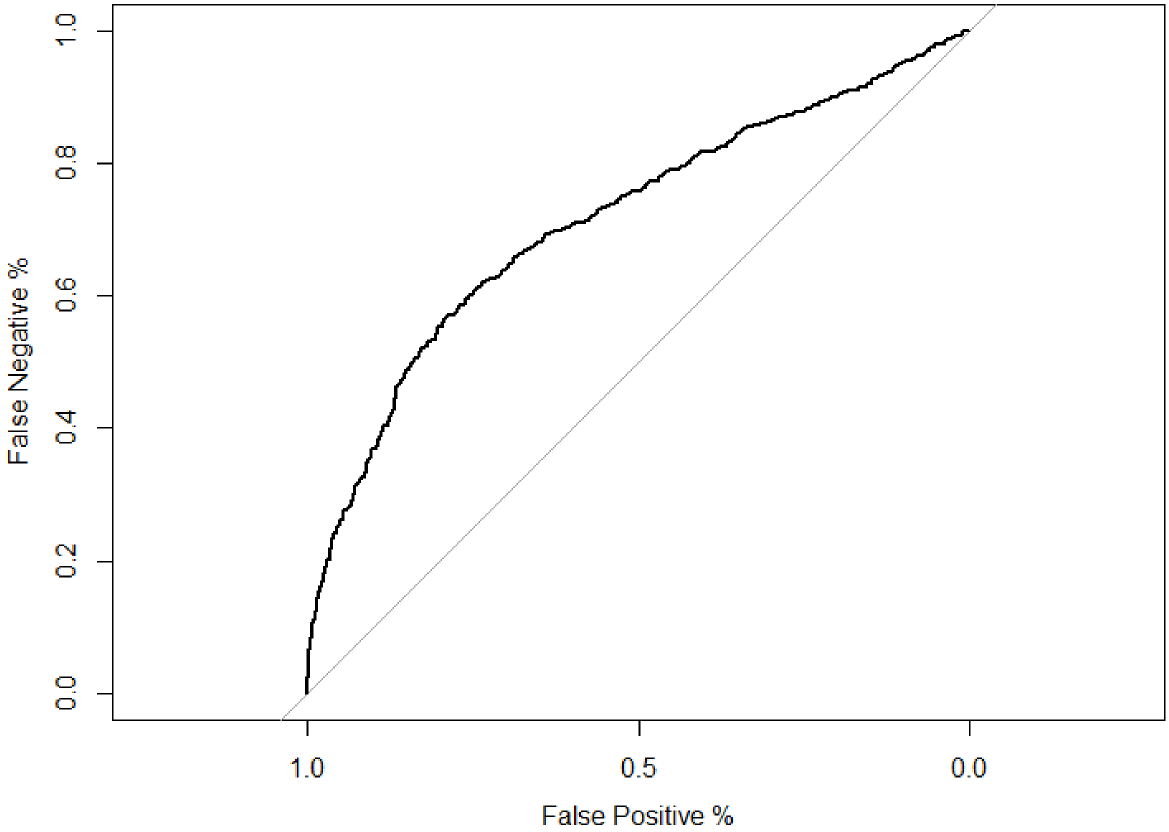
9. Based on the residuals plots from Exercises 6 - 8, is the linearity assumption satisfied? Briefly explain why or why not.

The linearity assumption is satisfied because the quantitative predictor plot is somewhat random, even thought there is a clear divide into two parts, and the categorical predictor is normal because the means of each type of key are close to 0.

Part III: Model Assessment & Prediction

10. Plot the ROC curve and calculate the area under the curve (AUC). Display at least 5 thresholds (n.cut = 5) on the ROC.

```
roc_curve <- roc(spotify_aug, target, .fitted, plot=TRUE, xlab="False Positive %",
                 ylab="False Negative %" )
roc_curve$auc
```



Plot:

AUC:

```
Area under the curve: 0.7138
```

11. Based on the ROC curve and AUC in the previous exercise, do you think this model effectively differentiates between the songs the user likes versus those he doesn't?

Based on the curves, I believe that the model does not effectively differentiate between the songs the user likes versus those they do not because the false positive and false negative rate, even when an optimal threshold is chosen, is around 30-50% which is really just as good as a random guess.

12. You are part of the data science team at Spotify, and your model will be used to make song recommendations to users. The goal is to recommend songs the user has a high probability of liking.

- Choose a threshold value to distinguish between songs the user will like and those the user won't like. What is your threshold value? Use the ROC curve to help justify your choice.

Using the following code to calculate the optimal threshold: `coords(roc_curve, "best", ret = "threshold")` I have found that the optimal threshold is 0.2066671, so around 0.2.

13. Make the confusion matrix using the threshold chosen in the previous question.

```
spotify_aug <- mutate(spotify_aug, pred = ifelse(.fitted > 0.2066671, 1, 0))

spotify_aug$pred <- as.factor(spotify_aug$pred)

confusionMatrix(spotify_aug$pred, spotify_aug$target)
```

```
          Reference
Prediction   0   1
         0 790 442
         1 207 578
```

14. Use the confusion matrix from the previous question to answer the following:

These questions were answered using information given by the code in the previous question.

- What is the proportion of true positives (sensitivity)?

  0.7924

- What is the proportion of false positives (1 - specificity)?

  1 - 0.5667 = 0.4333

- What is the misclassification rate?

  1 - 0.6782 = 0.3218