DDoS Categorical Data Analysis Project

Firass Elhouat

2025-04-04

Contents

1 Introduction

In today's digital landscape, Distributed Denial-of-Service (DDoS) attacks are among the most disruptive and costly cybersecurity threats, impacting organizations across various sectors — from government agencies and educational institutions to healthcare providers and private enterprises. These attacks aim to overwhelm a target's network infrastructure with illegitimate traffic, making online services inaccessible to legitimate users and critical stakeholders.The financial implications of DDoS attacks are significant. According to a 2023 insight report by Zayo Group, the average attack lasts around 68 minutes and costs approximately $408,000 — factoring in revenue loss, detection, recovery, and mitigation costs, (Zayo Group, 2024).

Beyond immediate disruptions, DDoS attacks carry long-term consequences such as reputation damage, legal ramifications, operational delays, and customer attrition. A notable example is the 2011 PlayStation Network (PSN) attack, which rendered Sony's services inaccessible for nearly a month. The incident not only resulted in substantial financial losses but also drove users to competitors like Steam and Xbox Live, ultimately tarnishing Sony's brand reputation, (Garcia, 2021).Given the growing sophistication and impact of such attacks, this project focuses on developing a logistic regression model to predict and classify DDoS activity based on network traffic features. The goal is to identify significant predictors of malicious traffic and build a model that supports real-time threat detection and mitigation strategies. The data used for this analysis and modeling building is the CIC-DDoS2019, the exact structure of the dataset and the column attributes is found in the appendix in table 1 and table 2.

2 Data Preprocessing

To prepare the CIC-DDoS2019 dataset for analysis and modeling, the response variable label was converted to binary format,(DDoS → 1, BENIGN → 0). Thirteen categorical variables were converted to their respective structures as.factor. Furthermore, we checked for possible null values, in this case there were non, however the quality of the data was questionable, as a number of variables contained invalid entries. For instance, flow_packets_s and flow_bytes_s contained 85 invalid entries, which were removed before converting the columns to numeric structure. In addition, six bulk-related variables displayed either zero variance or a large amount of zero values across observations and were excluded due to their lack of informativeness.Furthermore, redundant variables were also identified, specifically avg_fwd_segment_size duplicated fwd_packet_length_mean, and avg_bwd_segment_size duplicated bwd_packet_length_mean. These duplicates were dropped to reduce potential complications in terms of them being highly redundant. In terms of the categorical variables, a number of them displayed high imbalance, particularly "protocol", "flag counts variables" and other TCP control flags (PSH/URG) across FWD/BWD directions. These variables exhibited skewed or incomplete representation, especially under the DDoS class. As a result, most were removed, expect for ack_flag_count and psh_flag_count, which remained balanced and well-distributed.

Another insight gained from the exploratory data analysis, was that most of the numeric variables were highly right-skewed, indicating that the majority of the observations were concentrated around lower values. This distribution posed challenges during model fitting, as the model often failed to converge and also triggered the warning : "Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred". This indicated perfect separation due to the non transformed variables. To address this issue, I firstly had to remove values that fell outside the 2.5 and 97.5 percentile range, and found that applying a log or log1p transformation to all the numeric variables helped not only reduce skewness but also stabilize our models. In figure 1.1 we can see how the distributions are before and after using a log or log1p transformation. These transformations not only significantly improved our model, but further highlighted relationships between variables. Furthermore, in the model development I decided to us a sample of the data, as this proved to be beneficial, especially with computational efficiency, and model fitting.

2.1 Correlation Matrix and Variable Selection

During variable selection, we removed variables with high correlation (above 0.7) to reduce multicollinearity. However, some of these variables contained useful information, so instead of discarding them entirely, we treated them as alternatives in separate model runs. Correlated pairs were never included in the same model; rather, we evaluated each version using criteria such as AIC values, and goodness of fit test. This approach allowed us to retain informative variable without compromising model stability. The final dataset was narrowed down to 16 explanatory variables and 1 response variable, a significant reduction from the original 85 columns. The correlations matrix heat maps clearly show extreme correlations between a variables, thus included the before and after variable selection in figure 2.1 and 2.2.

3 Model Development

During the model development stage, it was essential to identify the most effective modeling approach for this specific dataset. This required a structured process grounded in exploratory analysis, data refinement, and iterative model evaluation. As stated before, as numeric variables with highly skewed and extreme values needed to be handled properly. If left unaddressed, these characteristics distorted coefficient estimates, inflated standard errors, and comprise overall model performance. Furthermore, although we reduced the number of highly correlated variables, we also explored different models using various subset of predictors to identify the best-fitting model based on specific criteria. In this case, I fitted three initial models and then applied a stepwise selection procedure, which iteratively adds or removes variables based on the Akaike Information Criterion (AIC) to optimize model performance.

In this case, i fitted three initial models and then applied a stepwise selection procedure, which iteratively added and removed variables based on the Akaike Information Criterion (AIC), which in the end provides us with a set of predictors that provide the most importance, and removing ones that increases the AIC value. For clarity, these are the three initial models with their respective variables and transformation effects used;

IntialModel1 contains these predictors: psh_flag_count + down_up_ratio + flow_packets_s + fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + fwd_iat_total + idle_mean + total_backward_packets

IntialModel2 contains these predictors: ack_flag_count + down_up_ratio + flow_packets_s + fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + fwd_iat_total + min_packet_length + idle_mean

IntialModel3 contains these predictors: down_up_ratio + idle_mean + flow_packets_s + psh_flag_count + fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets + bwd_header_length

This strategy minimized the risk of omitting potentially informative predictors due to their correlation-induced exclusion within any single model. By diversifying the variable combinations across models, we aimed to comprehensively explore this. Furthermore, certain variables were deliberately excluded despite their potential usefulness, due to their tendency to induce perfect separation. These included, min_seg_size_forward, average_packet_size, and total_length_of_fwd_packets. This modeling strategy supports a more robust selection process by ensuring model stability, avoiding overfitting, while also account for potential individual predictors across different configurations.

Next, we applied the stepAIC function to perform further variable selection. In our case, the function eliminated down_up_ratio from Model 1 and flow_iat_min from Model 2, while Model 3 remained unchanged. This outcome reinforces the robustness of our initial manual variable selection approach, as only a few variables were removed. The AIC values from the initial models are as follows: Model 1 [6484.918], Model 2 [6501.771], and Model 3 - [5459.524]. Based on these results, Model 3 is clearly the most favorable, exhibiting the lowest AIC values.

3.1 Exploring Interaction Effects

In the subsequent phase of the model development, I explored potential interaction effects to asses whether the final model which in this case Model 3, could further be improved prior to conducting diagnostics and hypothesis tests. as interaction effects may enhance the model by capturing the conditional influence that one predictor may have depending on the level of another. In essence, these effects allow us to represent situations in which the relationship between a predictor and the response is not constant but varies with another predictor.

In this case, the initial model 3 was refitted with a number of interaction effects, thus the new model is;

Model 3: Includes the original 10 predictors plus and 6 interaction effects

Main Effects: down_up_ratio + idle_mean + flow_packets_s + psh_flag_count + fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets + bwd_header_length

Interaction Effects: - bwd_iat_min:down_up_ratio - flow_packets_s:psh_flag_count - idle_mean:bwd_iat_min - idle_mean:flow_packets_s - psh_flag_count:flow_iat_min - flow_iat_min:idle_std

In this case, Model 3 with interaction effects achieved an AIC value of 4239.854, representing a substantial improvement over the model without interaction effects, which had an AIC value of 5459.524. This suggests that incorporating interaction terms improves the model, we can see the model summary in the appendix in pages 10 and 11 in the appendix section. With the model equation defined as

$log(\frac{P(DDoS)}{1-P(DDoS)})$ = β_0 + β_1×down_up_ratio + β_2×idle_mean β_3×flow_packets_s + β_4×psh_flag_count1
+ β_5×fwd_iat_min + β_6×flow_iat_min + $\beta_7$idle_std + β_8×bwd_iat_min + β_9×subflow_fwd_packets +
β_{10}×bwd_header_length + β_{11}×down_up_ratio:bwd_iat_min + β_{12}×flow_packets_s:psh_flag_count1 + β_{13}×idle_mean:bwd_iat_min
+ β_{14}×idle_mean:flow_packets_s + β_{15}×psh_flag_count1:flow_iat_min + β_{16}×flow_iat_min:idle_std

However, it is important to note that while AIC is useful for model comparison, it does not measure the goodness-of-fit directly, nor does it test the statistical significance of the model. Therefore, in the next section, we will proceed with diagnostics checks, checking residuals, and conducting formal hypothesis testing to further validate our final model. This includes overall model significance test, and partial tests for interaction effects.

4 Model Evaluation & Diagnostics

In this section, the primary objectives is to thoroughly evaluate the final model by conducting a comprehensive set of diagnostics. This includes assessing multicollinearity through the Variance Inflation Factor (VIF), examining the residuals, and evaluating overall model fit. These steps are essential before the hypothesis tests, as they ensure that the model meets key statistical assumptions, provides stable coefficient estimates, and yields reliable inference and predictions.

4.1 Multicollinearity Diagnostics

In terms of our final model's variance inflation factor (VIF) values, the model without interaction effects show values ranging from approximately 1.47 to 3.78, as shown in the table 3 for the multicollinearity diagnostics. With the inclusion of interaction terms, an increase in VIF values is expected and inevitable due to the added complexity. One common approach to mitigate this issue is by centering the variables. Despite this, in our final model, none of the VIF values exceed 7.91, which remains within an acceptable range, suggesting this does not pose any serious concerns, the full VIF tables around find in the appendix, pages 11 and 12.

4.2 Residual Diagnostics

Based on the residuals summary of our final model and plots 3.1 and 3.2, it shows that the residuals are reasonably centered around zero, with a mean of 0.08 and a median 0.16, further more a min of -3.62067, and a max of 2.97580. However, these values alone do not provide sufficient insight into the overall fit of the model. To gain further insight, we could use a binned residuals plot, which groups residuals based on estimated probability of our response variable and helps identify patterns or areas where the model may not fit well. In this case, we used the pearson residuals, with 10 bins that it divides the data into. As a result, we can see in plot 3.3, that 90% of the residuals are inside the error bounds, which may suggest that the model fits the data well.

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.62067 -0.01945  0.16140  0.08021  0.35259  2.97580
```

4.3 Goodness-Of-Fit Test

A more formal and widely used approach to assess the fit of our logistic regression model is through the use of a Hosmer and Lemeshow goodness-of-fit test. This test compares the observed and expected frequencies of the response variables within groups. For the hypothesis testing, we use an alpha α level of 0.05. The null and alternative hypotheses are:

$$H_0 : \text{The model does not fit the Data}, \ H_a : \text{The model fits the data}$$

The results of the test were as followed;

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  as.numeric(as.character(SampledData$label)), fitted(MODEL3)
## X-squared = 12.554, df = 8, p-value = 0.1281
```

Given that the p-value [0.1281] is greater than 0.05, we state that we fail to reject the null hypothesis H_0, and conclude that there is no significant evidence to suggest a lack of fit in our model. This implies that our logistic regression model provides an adequate fit to the observed data. As our final model as been fully evaluated, we can processed with other hypothesis tests.

4.4 Hypothesis Tests

In this section, we aim to conduct formal hypothesis tests to further evaluate our model's performance. Given that our final model does not indicate a lack of fit, we can confidently proceed with these tests to assess the statistical significance of our findings and ensure the robustness of our model. This is achieved by using Likelihood Ratio Tests, which compares nested models by examining the difference in their deviance. The test statistic follows a chi-squared distribution with k degrees of freedom, where k is the difference in the number of parameters between the two models being compared (e.g, the number of predictors added in the full model compared to the null model). The corresponding p-value helps determine the significance of adding those predictors in our model. The likelihood ratio test is conducted this way: $\chi^2 =$ Null Devience - Residual Devience, p-value = (χ^2_{k-1}). The hypothesis tests which have been conducted in this paper is as followed;

1. Performing a formal hypothesis test to determine if our logistic regression model is statistically significant.

$$H_0 : \beta_j = 0 : j = 1 \cdots 16 H_a : \exists\, \beta_j \neq 0 : j = 1 \cdots 16$$

From this test, the results are $\chi^2 = 7580.7$, df = 16, $p = 2.2e - 16$, Thus, we reject the null hypothesis H_0 and conclude that this logistic regression model, which includes 10 predictors and 6 interaction effect, is statistically statistically significant in predicting the type of network traffic (DDoS or Benign)

2. Performing a formal hypothesis test to determine if adding subflow_fwd_packets to a logistic regression model that already includes 9 other predictors and 6 interaction effects.

   $$H_0 : \beta_j = 0 : j = 9 H_a : \exists \beta_j \neq 0 : j = 9$$

   From this test, the results are $\chi^2 = 250.92$, df = 1, $p = 2.2e - 16$. Thus, we reject the null hypothesis H_0 and conclude that adding subflow_fwd_packets to a model that already includes the 9 predictors and 6 interaction effects, is statistically statistically significant in predicting the type of network traffic (DDoS or Benign)

3. Performing a formal hypothesis test to determine if adding bwd_header_length to a logistic regression model that already includes 9 other predictors and 6 interaction effects is statistically significant.

   $$H_0 : \beta_j = 0 : j = 10, H_0 : \exists \beta_j \neq 0 : j = 10$$

   From this test, the results are $\chi^2 = 1126.2$, df = 1, $p = 2.2e - 16$. Thus, we reject the null hypothesis H_0 and conclude that adding bwd_header_length to a model that includes the 9 predictors and 6 interaction effect, is statistically statistically significant in predicting the type of network traffic (DDoS or Benign)

4. Performing a formal hypothesis test to determine if a more complex model is statistically significant. In this case, we are testing if adding the 6 interaction effects to a model that already includes 10 predictors is is statistically significant.

   $$H_0 : \beta_j = 0 : j = 11, 12, 13, 14, 15, 16 H_a : \exists \beta_j \neq 0 : j = 11, 12, 13, 14, 15, 16$$

   From this test, the results are $\chi^2 = 1231.7, p = 2.2e - 16$. Thus, we reject the null hypothesis H_0 and conclude that adding complexity to this model through interaction effects, is statistically significant in predicting the the type of network traffic (DDoS or Benign).

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

4.5 Model Predictive Power Summary:

Based on the model's predictive power, it yields very promising results. The confusion matrix indicates that the model correctly predicted 7,099 true positives (DDoS attacks) and 2,251 true negatives (Benign traffic), with 511 false negatives, and 139 false positives. The model has an overall accuracy of 93.5%, with a 95% confidence interval, it indicates that the accuracy lies between 93% and 93.98%. The sensitivity of the model is 98.08% meaning it correctly identifies a high proportion of DDoS traffic. The specificity, is at 81.5%, suggesting that there is some room for improvement in reducing false positives, where begin traffic is incorrectly classified as DDoS.

Despite this, the model's predictive performance remains strong overall, without signs of poor performance or overfitting. Furthermore, the Kappa score of 0.8304 indicates strong agreement between the model's predictions and the true labels, and not by chance. Furthermore, in plot 4.1, of the Receiver Operating Characteristic (ROC) curve, it has an Area Under the Curve (AUC) value of 0.955, indicating an excellent model performance in distinguishing between the two classes. An AUC of higher then 0.5, means that it is not just randomly guessing, but demonstrates that the model has a high ability to discriminate between DDoS traffic and Benign Traffic, further validating its effectiveness in a real-world application.

5 Conclusion & Discussion

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

Before concluding this project, our initial aim was to interpret some of these coefficients and how do they translate to real world conditions. In this case, our model equation is defined as;

$logit(\hat{p_i})$ = 9.248169 $-2.596762\times$down_up_ratio
0.083959$\times$idle_mean $-1.063623\times$flow_packets_s $+1.076716\times$psh_flag_count1 $+0.285898\times$fwd_iat_min
$-0.642793\times$flow_iat_min $+0.002782$idle_std $+0.026213\times$bwd_iat_min
$-2.103303\times$subflow_fwd_packets $-1.443926\times$bwd_header_length
$+0.468262\times$down_up_ratio:bwd_iat_min
$+1.302090\times$flow_packets_s:psh_flag_count1
$+0.034374\times$idle_mean:bwd_iat_min + $-0.074022\times$idle_mean:flow_packets_s
$+0.487390\times$psh_flag_count1:flow_iat_min $+0.055873\times$flow_iat_min:idle_std

In this case, we will interpret five coefficients and construct their respective confidence intervals using the profile likelihood method.

Main Effects Interpretations and Confidence Interval:

flow_packets_s (log transformed): Holding all other variables constant, as flow_packets_s increases by 1%, (since it is log-transformed), we divide the coefficient -1.063623 by 100 to get -0.01063623. Thus, the expected odds of DDoS decreases by a factor of exp(-0.01063623) = 0.9894201. This tells us that for every 1% increase in the flow packets sent per seconds the odds of a DDoS decreases by 1.06%. Furthermore, we are 95% confident that the expected odds is between exp(-1.16406007/100) = 0.9884269 and exp(-0.96931508/100) = 0.9903537, when holding all other variables constant.

psh_flag_count1: Holding all other variables constant, if a packet is sent with a PUSH indicator, the expected odds of a network traffic being a DDoS attack increased by a factor exp(1.076716) = 2.933. This means that relative to packets with without the PUSH indication (0), PUSH Packets (1) are associated with approximately 2.93 times higher odds of the traffic being a DDoS attack. Using profile confidence interval, we are 95% confident that the expected odds is between exp(0.61611548) = 1.851721 and exp(1.53926111) = 4.661145, when holding all other variables constant.

flow_iat_min (Minimum time between two packets sent in the flow): Holding all other variables constant, as the minimum time between two packets in the flow increases by one unit, the expected odds of network traffic being a

DDoS attack decreases by a factor of exp(-0.642793) = 0.526. This indicates that longer time games between packets in the flow are associated with lower odds of DDoS traffic. With our confident interval output, we are 95% confident that the expected odds are between exp(-0.74886243) = 0.4729042, and exp(-0.54086540) = 0.5822442

Interaction Effects Interpretations and Confidence Interval:

psh_flag_count1:flow_iat_min: Holding all other variables constant, for packets with PUSH flag relative to those without it, each one unit increase in the minimum inter-arrival time between flow packets increases the expected odds of DDoS by a factor of exp(0.487390) = 1.628. This implies that PUSH flag combined with wider packet timing is more indicative of DDoS traffic. Using profile confidence interval, we are 95% confident that the expected odds is between exp(0.35557301) = 1.426998 and exp(0.62019509) = 1.859291, when holding all other variables constant.

flow_iat_min:idle_std: Holding all other variables constant, as the interaction between the minimum flow inter-arrival time and the standard deviation of idle time increases by one unit, the expected odds of DDoS increase by a factor of exp(0.055873) = 1.057. This means that flows which exhibit both high variability in idle time and larger gaps between packets are slightly more likely to be DDoS related, potentially reflecting bursts of traffic typical of attack patterns. Furthermore, we are 95% confident that the expected odds is between **exp(**0.02377912**) =** 1.024064 and **exp(**0.09498267**) =** 1.09964, when holding all other variables constant.

5.1 Conclusion

This study set out to distinguish between DDoS and normal network traffic by building and interpreting a logistic regression model using flow-based numeric variables, and flag indicator variables. The final model demonstrates a strong predictive power and interpretability, offering valuable insights into the behavioral patterns associated with DDoS attacks. For main effects, we observed that certain traffic characteristics, such as high packet rates per second, and longer flow inter-arrival times are generally associated with lower odds of DDoS attacks. Conversely, variables like presence of PUSH flags, higher variability in idle time, and longer gaps between packets tend to increase the likelihood of traffic being malicious.

The model also highlighted important interaction effects, revealing that combinations of variables such as PUSH flags paired with wider flow inter-arrival times are significantly indicative of DDoS attacks. These interactions uncover more nuanced and complex behaviors in malicious traffic that individual variables alone may not capture. By integrating several hypothesis test and model evaluation techniques, this study establishes a solid foundation for enhancing early detection systems and advancing research in network traffic analysis, particularly in distinguishing key packet characteristics that separate normal traffic from DDoS attacks.

Future Work Looking ahead, future work could involve exploring more diverse dataset that include a multiclass response variable such as distinguishing between various other types of network malwares, phishing, and even different categories of DDoS. This would allow for a more granular classification and better reflect real-world conditions where network traffic may involve multiple forms of malicious attacks. Additionally, the addition of working with more advanced modeling techniques, such as ensemble method, or deep learning architectures, could further improve the predictive performance and offer deeper insights into the vast complex traffic patterns.

6 Reference List & GitRepo

- Github Project Repo: https://github.com/FirassEL/STAT-401-Project-DDOS-ATTACK-DATA.git

- Zayo Group (2024) *Average ddos attack cost businesses nearly half a million dollars in 2023, according to New Zayo Data: Press release: Zayo*, *Zayo.com*. Available at: https://www.zayo.com/newsroom/average-ddos-attack-cost-businesses-nearly-half-a-million-dollars-in-2023-according-to-new-zayo-data/ (Accessed: 18 March 2025).

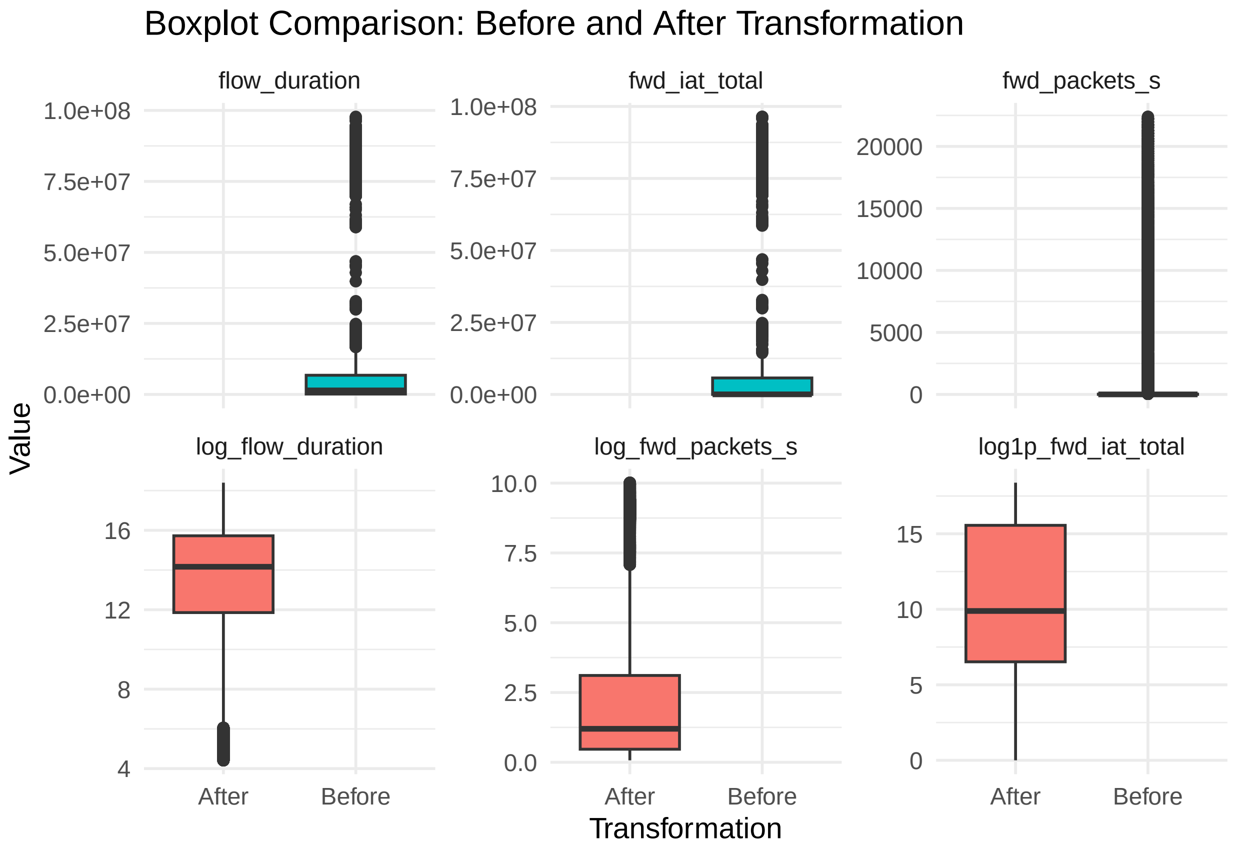
- Garcia, D.M. (2021) *The 2011 PlayStation Network Hack – what actually happened?*, *WestSide Story*. Available at: https://wsswired.com/4837/entertainment-3/the-2011-playstation-network-hack-what-actually-happened/ (Accessed: 18 March 2025).

- CIC_DDOS2019 Description details source. https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt (Accessed: 1 March 2025).

- CIC_DDoS2019 Sourced: https://www.unb.ca/cic/datasets/index.html (Accessed: 1 March 2025).

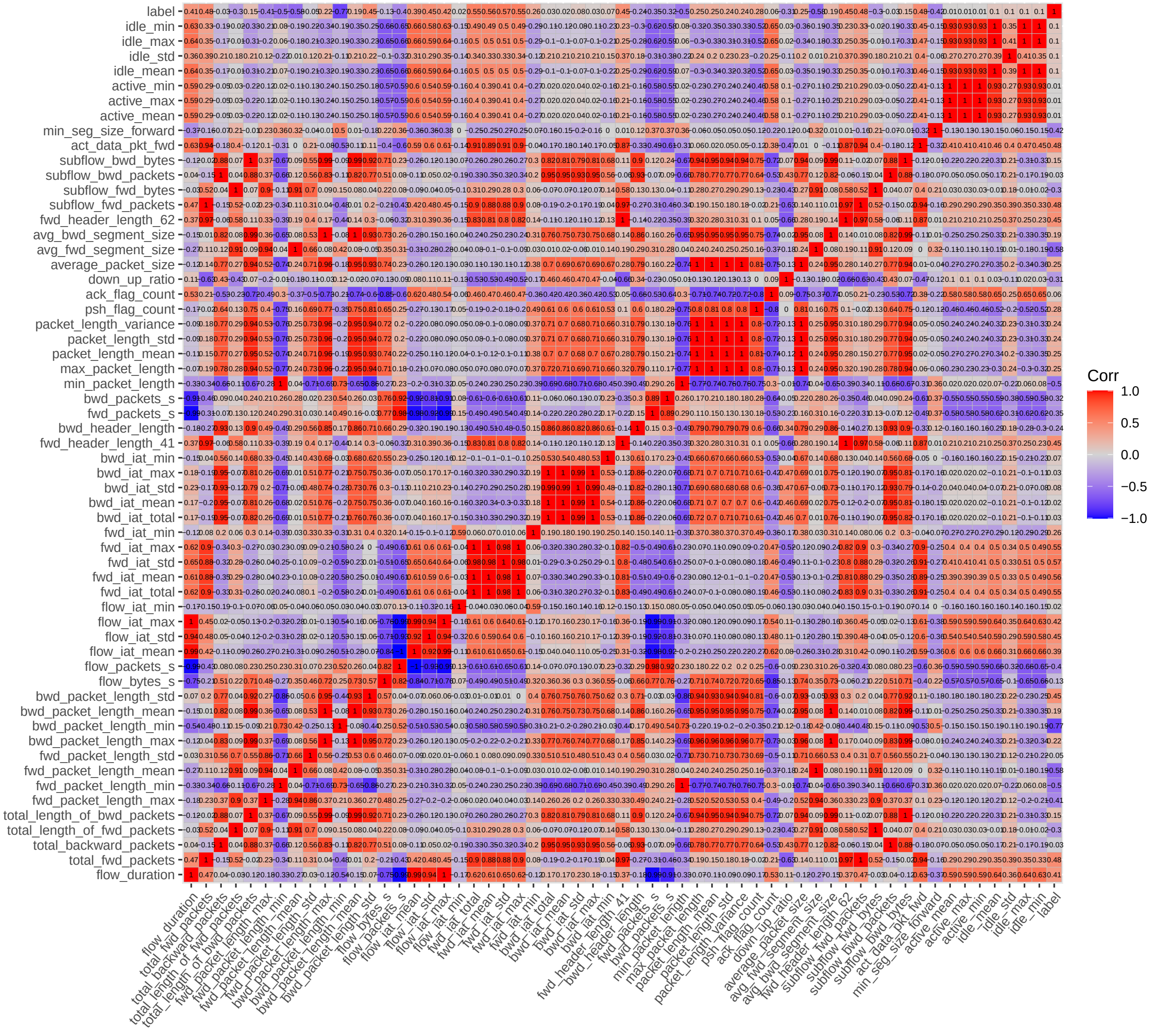
7 Appendix

Plot 1.1 Distribution changes before and after log transformation



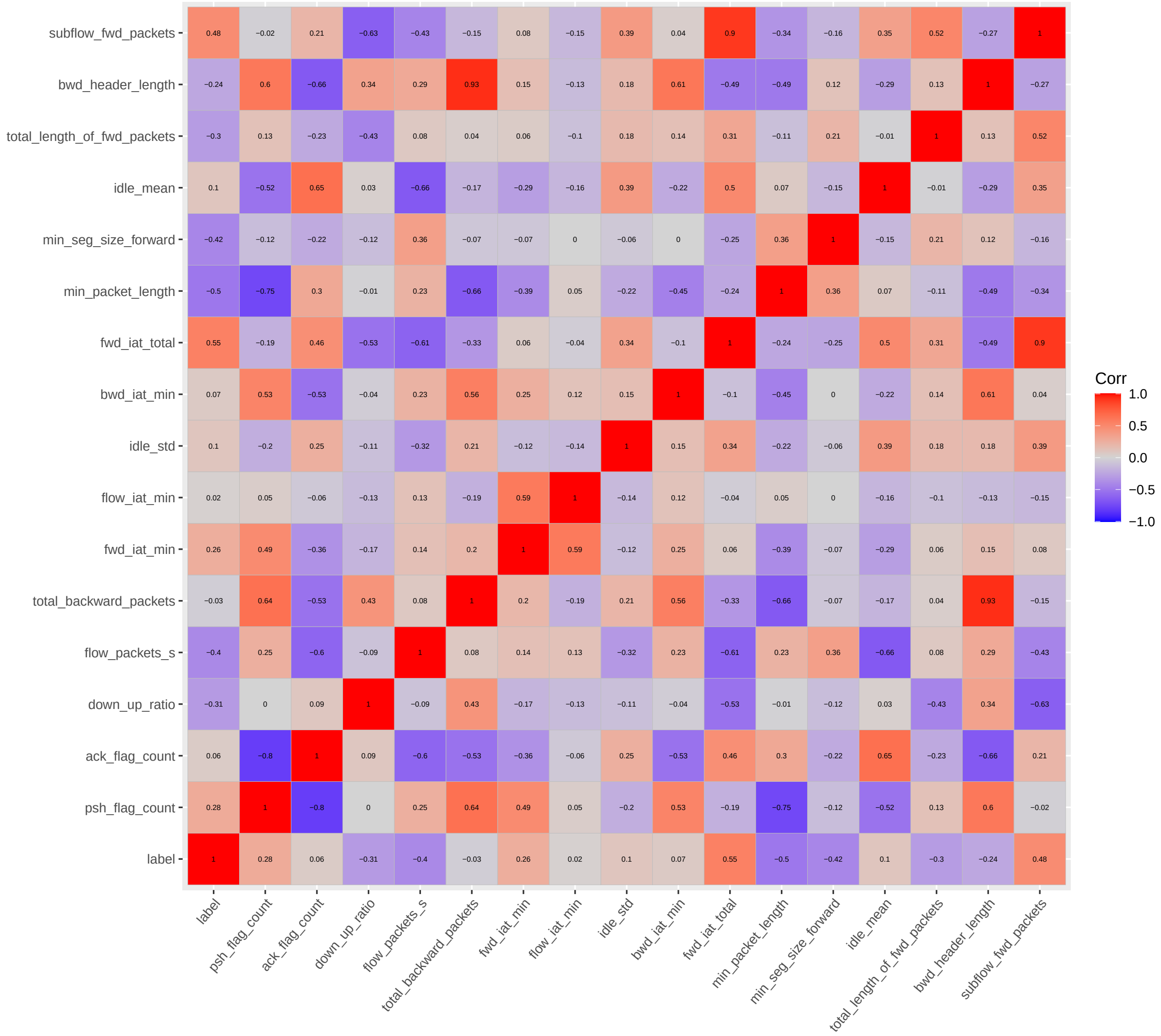
Plot 2.1 Correlation matrix full plot

Figure 2.1: Correlation Matrix Heat Map Of All Variables



Plot 2.2 Correlation matrix full plot

Figure 2.2: Correlation Matrix Heat Map Of Manually Selected Variables



7.1 Final Model Output

```
# final model output summary
summary(MODEL3)
```

```
##
## Call:
## glm(formula = label ~ down_up_ratio + idle_mean + flow_packets_s +
##     psh_flag_count + fwd_iat_min + flow_iat_min + idle_std +
##     bwd_iat_min + subflow_fwd_packets + bwd_header_length + bwd_iat_min:down_up_ratio +
##     flow_packets_s:psh_flag_count + idle_mean:bwd_iat_min + idle_mean:flow_packets_s +
##     psh_flag_count:flow_iat_min + flow_iat_min:idle_std, family = binomial(link = "logi
##     data = FinalData, control = glm.control(maxit = 1000))
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     9.248169   0.356713  25.926  < 2e-16 ***
## down_up_ratio                  -2.596762   0.180536 -14.384  < 2e-16 ***
## idle_mean                      -0.083959   0.012852  -6.533 6.46e-11 ***
## flow_packets_s                 -1.063623   0.049668 -21.415  < 2e-16 ***
## psh_flag_count1                 1.076716   0.235406   4.574 4.79e-06 ***
## fwd_iat_min                     0.285898   0.032252   8.865  < 2e-16 ***
## flow_iat_min                   -0.642793   0.053077 -12.111  < 2e-16 ***
## idle_std                        0.002782   0.015627   0.178  0.85871
## bwd_iat_min                     0.026213   0.043699   0.600  0.54860
## subflow_fwd_packets            -2.103303   0.138421 -15.195  < 2e-16 ***
## bwd_header_length              -1.443926   0.056360 -25.620  < 2e-16 ***
## down_up_ratio:bwd_iat_min       0.468262   0.052092   8.989  < 2e-16 ***
## flow_packets_s:psh_flag_count1  1.302090   0.063042  20.654  < 2e-16 ***
## idle_mean:bwd_iat_min           0.034374   0.003835   8.963  < 2e-16 ***
## idle_mean:flow_packets_s       -0.074022   0.008635  -8.572  < 2e-16 ***
## psh_flag_count1:flow_iat_min    0.487390   0.067474   7.223 5.07e-13 ***
## flow_iat_min:idle_std           0.055873   0.018107   3.086  0.00203 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11786.6  on 9999  degrees of freedom
## Residual deviance:  4205.9  on 9983  degrees of freedom
## AIC: 4239.9
##
## Number of Fisher Scoring iterations: 7
```

Multicollinearity Output for Model 3 without interaction effects

```
# Multicollinearity Check for base model
performance::multicollinearity(InitialModel3)
```

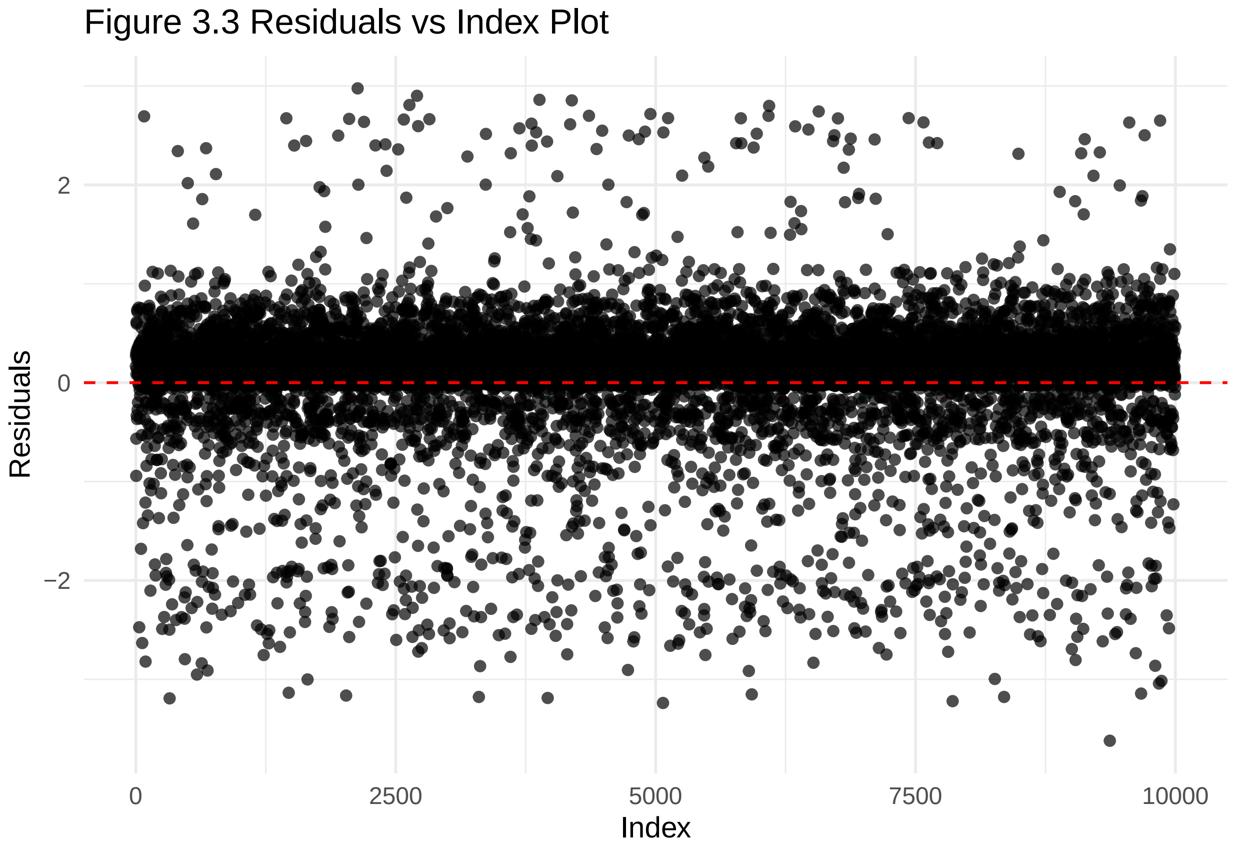
```
## # Check for Multicollinearity
##
## Low Correlation
##
##                 Term  VIF   VIF 95% CI Increased SE Tolerance Tolerance 95% CI
##        down_up_ratio 3.68 [3.55, 3.80]         1.92      0.27     [0.26, 0.28]
##            idle_mean 3.48 [3.37, 3.60]         1.87      0.29     [0.28, 0.30]
##       flow_packets_s 2.60 [2.52, 2.69]         1.61      0.38     [0.37, 0.40]
##       psh_flag_count 3.68 [3.56, 3.81]         1.92      0.27     [0.26, 0.28]
##          fwd_iat_min 3.38 [3.27, 3.50]         1.84      0.30     [0.29, 0.31]
##         flow_iat_min 3.78 [3.65, 3.91]         1.94      0.26     [0.26, 0.27]
##             idle_std 1.47 [1.43, 1.51]         1.21      0.68     [0.66, 0.70]
##          bwd_iat_min 2.11 [2.05, 2.17]         1.45      0.47     [0.46, 0.49]
##  subflow_fwd_packets 3.72 [3.59, 3.84]         1.93      0.27     [0.26, 0.28]
##    bwd_header_length 3.76 [3.63, 3.89]         1.94      0.27     [0.26, 0.28]
```

Multicollinearity Output for Model 3 with interaction effects

```r
# Multicollinearity Check for final model
performance::multicollinearity(MODEL3)
```

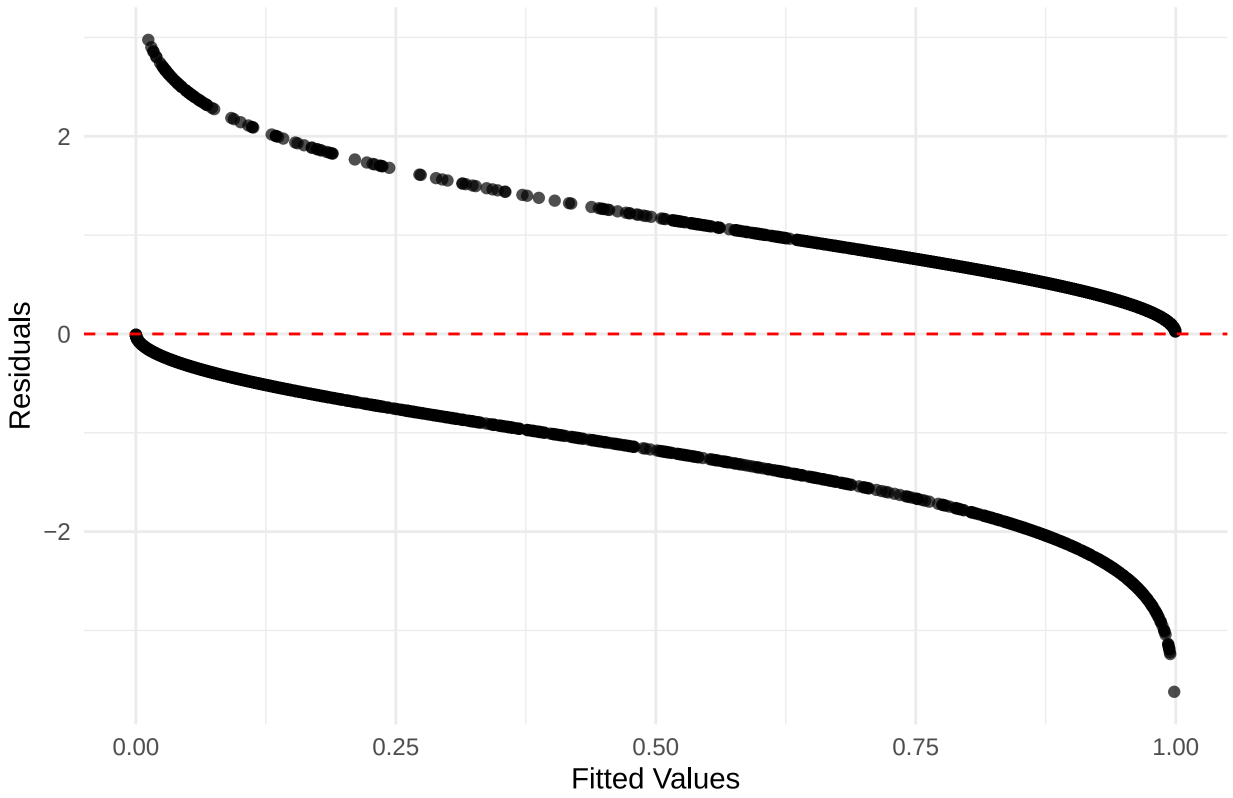
```
## # Check for Multicollinearity
##
## Low Correlation
##
##                          Term  VIF   VIF 95% CI Increased SE Tolerance
##                   fwd_iat_min 3.24 [3.14, 3.35]         1.80      0.31
##                  flow_iat_min 4.71 [4.55, 4.88]         2.17      0.21
##                      idle_std 2.72 [2.63, 2.80]         1.65      0.37
##                   bwd_iat_min 3.65 [3.53, 3.78]         1.91      0.27
##           subflow_fwd_packets 4.22 [4.08, 4.37]         2.05      0.24
##             bwd_header_length 4.01 [3.87, 4.15]         2.00      0.25
##     down_up_ratio:bwd_iat_min 2.70 [2.61, 2.79]         1.64      0.37
##         idle_mean:bwd_iat_min 3.26 [3.16, 3.37]         1.81      0.31
##      idle_mean:flow_packets_s 3.96 [3.83, 4.10]         1.99      0.25
##   psh_flag_count:flow_iat_min 4.29 [4.14, 4.44]         2.07      0.23
##         flow_iat_min:idle_std 1.29 [1.26, 1.32]         1.14      0.77
##  Tolerance 95% CI
##      [0.30, 0.32]
##      [0.20, 0.22]
##      [0.36, 0.38]
##      [0.26, 0.28]
##      [0.23, 0.25]
##      [0.24, 0.26]
##      [0.36, 0.38]
##      [0.30, 0.32]
##      [0.24, 0.26]
##      [0.23, 0.24]
##      [0.76, 0.79]
##
## Moderate Correlation
##
##                            Term  VIF   VIF 95% CI Increased SE Tolerance
##                   down_up_ratio 5.85 [5.65, 6.07]         2.42      0.17
##                       idle_mean 5.16 [4.98, 5.34]         2.27      0.19
##                  flow_packets_s 6.27 [6.05, 6.51]         2.50      0.16
##                  psh_flag_count 7.91 [7.63, 8.21]         2.81      0.13
##   flow_packets_s:psh_flag_count 6.83 [6.59, 7.08]         2.61      0.15
##  Tolerance 95% CI
##      [0.16, 0.18]
##      [0.19, 0.20]
##      [0.15, 0.17]
##      [0.12, 0.13]
##      [0.14, 0.15]
```

Plot 3.1: Residuals Index



Plot 3.2: Fitted vs Residuals plot

Figure 3.2 Fitted vs Residuals Plot

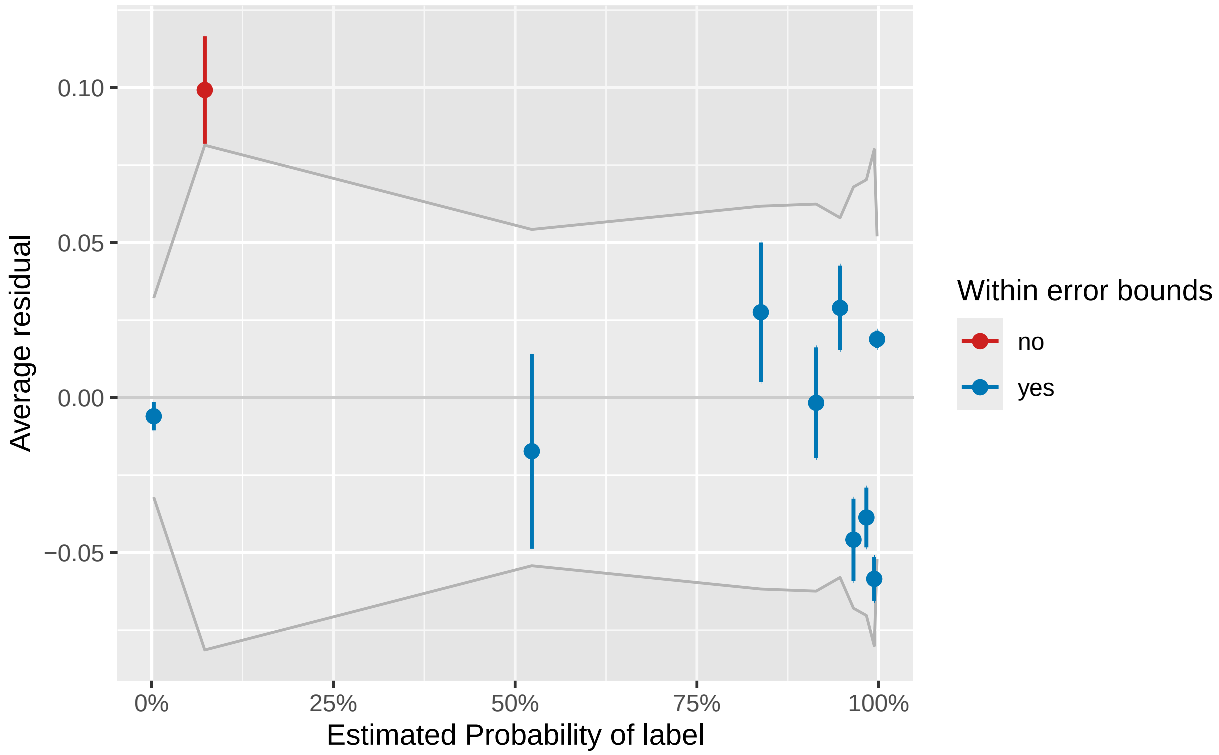


Plot 3.3: Binned Residual Plot

```
## Warning: About 90% of the residuals are inside the error bounds (~95% or higher would be
```

Figure 3.3 Binned Pearson Residuals Plot

Points should be within error bounds

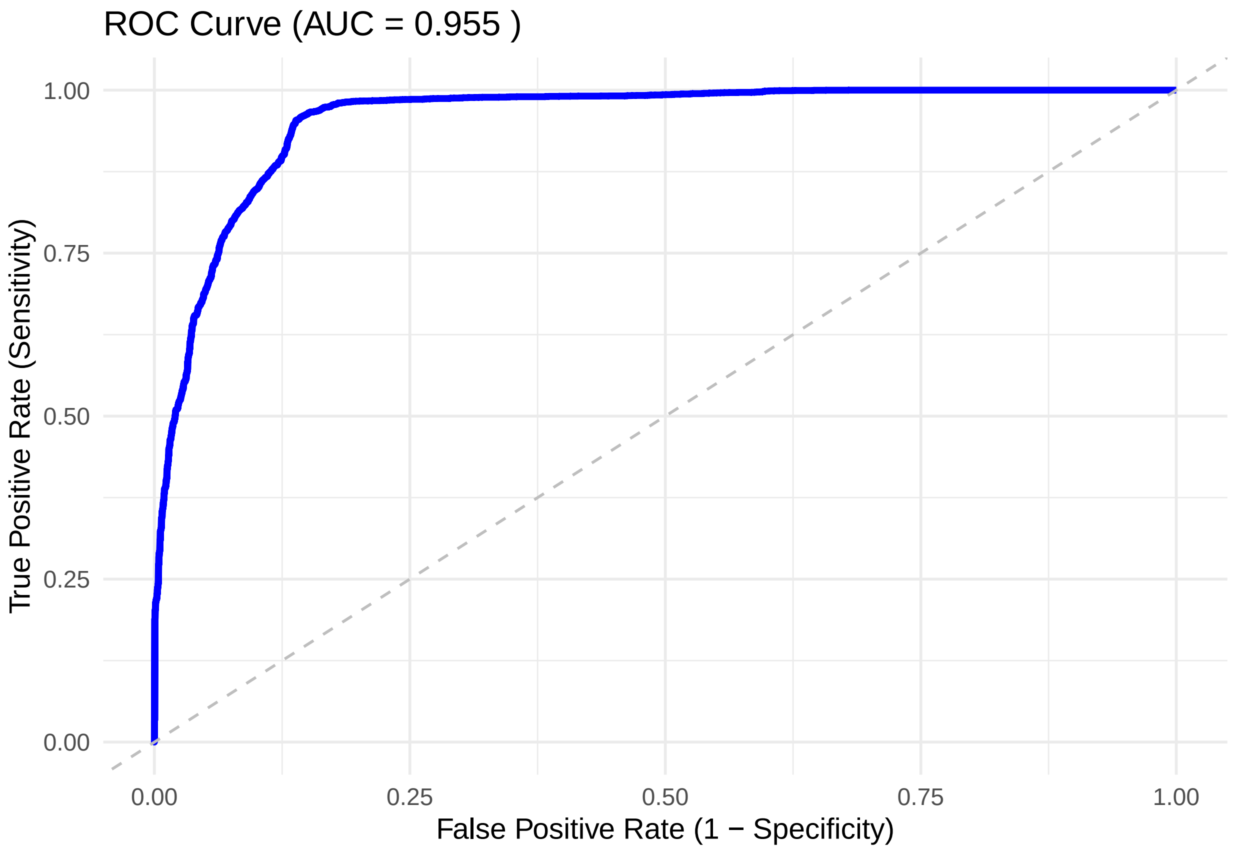


Plot 4.1: ROC Curve

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Table 1: CIC-DDoS2019 Dataset Structure and Class Distribution

Total number of records:	225,745
Total number of Variables (columns):	85
Total number of Continuous variables:	65
Total number of Categorical variables:	13
Instances of BENIGN (class 0):	97,718
Instances of DDoS (class 1):	128,027



7.2 Tables Of CIC-DDOS2019 DATASET DETAILS

```r
# --- loading libraries
library(tidyverse) # for extra functions
library(janitor) # for column name cleaning
library(ggcorrplot)# for correlation plot
library(ggnewscale) # for extra plot functions
library(car) # for VIF function
library(scales) # for extra plot functions
library(gridExtra) # extra plot functions (grid functions)
library(grid) # extra plot functions (grids)
library(kableExtra) # for tables functions
library(caret)
library(MASS)
```

Table 2: Appendix 1.1: CIC-DDoS2019 Dataset Attributes Details

Variable	Type	Details
flow_id	Identifier	A unqiue identifer assigned to each flow in the dataset.
source_ip	Identifier	The IP address of the device that originates the network traffic.
source_port	Identifier	The port number on the source device from which the network traffic is sent.
destination_ip	Identifier	The IP address of the device that received the network traffic.
destination_port	Identifier	The port number on the destination device that is used to receive the network traffic.
timestamp	Identifier	The time stamp at which the network flow occurred.
protocol	Categorical	Denotes the protocol used, with three levels; 0, 6 and 17
fin_flag_count	Categorical	The number of packets with FIN (Finish) flag, which signals the termination of a connection, with two levels, 1 and 0.
syn_flag_count	Categorical	The number of packets with SYN (Synchronize) flag, used to initiate a TCP connection, with two levels, 1 and 0.
rst_flag_count	Categorical	The number of packets with RST (Reset) flag, used to abruptly terminate a connection, with two levels, 1 and 0.
psh_flag_count	Categorical	The number of packets with PSH (Push) flag, which tells the receiver to process data immediately rather than buffering it, with two levels, 1 and 0.
ack_flag_count	Categorical	The number of packets with ACK (Acknowledgment) flag, which confirms the receipt data, with two levels, 1 and 0.
urg_flag_count	Categorical	The number of packets with URG (Urgent) flag, indicating that certain data should be processed immediately, with two levels, 1 and 0.
cwe_flag_count	Categorical	Will be renamed to correct name : CWR (Congestion window reduced) flag, used in TCP congestion control to signal a reduced congestion window, with two levels, 1 and 0.
ece_flag_count	Categorical	The number of packets with ECE (Explicit Congestion Notification Eco) flag, indicates network congestion, with two levels, 1 and 0.
fwd_psh_flags	Categorical	Number of time the PSH (Push) flag was set in packets travelling in the forward direction, with two levels, 1 and 0.
bwd_psh_flags	Categorical	Number of time the PSH (Push) flag was set in packets travelling in the backward direction, with two levels, 1 and 0.
fwd_urg_flags	Categorical	Number of time the URG (Urgent) flag was set in packets travelling in the forward direction, with two levels, 1 and 0.
bwd_urg_flags	Categorical	Number of time the URG (Urgent) flag was set in packets travelling in the forward direction, with two levels, 1 and 0.
flow_duration	Continuous	The duration of the flow in microseconds
total_fwd_packets	Continuous	The total packets in the forward direction
total_backward_packets	Continuous	The total packets in the backward direction
total_length_of_fwd_packets	Continuous	The total size of a packet in the forward direction
fwd_packet_length_max	Continuous	The total size of a packet in the backward direction
fwd_packet_length_min	Continuous	The maximum size of a packet in the forward direction.
fwd_packet_length_mean	Continuous	The minimum size of a packet in the forward direction.
fwd_packet_length_std	Continuous	The mean size of a packet in the forward direction.
bwd_packet_length_max	Continuous	The standard deviation size of a packet in the forward direction.
bwd_packet_length_min	Continuous	The maximum size of a packet in the backward direction.
bwd_packet_length_min	Continuous	The minimum size of a packet in the backward direction.
bwd_packet_length_mean	Continuous	The mean size of a packet in the backward direction.
bwd_packet_length_std	Continuous	The standard deviation size of a packet in the backward direction.
flow_bytes_s	Continuous	The number of flow bytes per second.
flow_packets_s	Continuous	The number of flow packets per second.
flow_iat_mean	Continuous	The average inter-arrival time between packets within a flow.
flow_iat_std	Continuous	The standard deviation inter-arrival time between packets within a flow.
flow_iat_max	Continuous	The maximum inter-arrival time between packets within a flow.
flow_iat_min	Continuous	The minimum inter-arrival time between packets within a flow.
fwd_iat_total	Continuous	The total inter-arrival time between packets sent in the forward direction.
fwd_iat_mean	Continuous	The average inter-arrival time between packets sent in the forward direction.
fwd_iat_std	Continuous	The standard deviation inter-arrival time between packets sent in the forward direction.
fwd_iat_max	Continuous	The maximum inter-arrival time between packets sent in the forward direction.
fwd_iat_min	Continuous	The minimum inter-arrival time between packets sent in the forward direction.
bwd_iat_total	Continuous	The total inter-arrival time between packets sent in the forward direction.
bwd_iat_mean	Continuous	The average inter-arrival time between packets sent in the forward direction.
bwd_iat_std	Continuous	The standard deviation inter-arrival time between packets sent in the forward. direction.
bwd_iat_max	Continuous	The maximum inter-arrival time between packets sent in the forward direction.
bwd_iat_min	Continuous	The minimum inter-arrival time between packets sent in the forward direction.
fwd_header_length_41	Continuous	The total bytes used for headers in the forward direction.
bwd_header_length	Continuous	The total bytes used for headers in the backward direction.
fwd_packets_s	Continuous	Number of forward packets transmitted per second.
bwd_packets_s	Continuous	Number of backward packets transmitted per second.
min_packet_length	Continuous	The minimum length of a packet.
max_packet_length	Continuous	The maximum length of a packet.
packet_length_mean	Continuous	The average length of a packet.
packet_length_std	Continuous	The standard deviation length of a packet.
packet_length_variance	Continuous	The variance length of a packet.
down_up_ratio	Continuous	Download and upload ratio.
average_packet_size	Continuous	The average size of a packet.
avg_fwd_segment_size	Continuous	The average size of data segments in the forward direction.
avg_bwd_segment_size	Continuous	The average size of data segments in the backward direction.
fwd_header_length_62	Continuous	Length of a packet header in the forward direction.
fwd_avg_bytes_bulk	Continuous	Average number of bulk bytes rate in the forward direction.
fwd_avg_packets_bulk	Continuous	Average number of bulk packets rate in the forward direction.
fwd_avg_bulk_rate	Continuous	Average number of bulk rate in the forward direction.
bwd_avg_bytes_bulk	Continuous	Average number of bulk bytes rate in the backward direction.
bwd_avg_packets_bulk	Continuous	Average number of bulk packets rate in the backward direction.
bwd_avg_bulk_rate	Continuous	Average number of bulk rate in the backward direction.
subflow_fwd_packets	Continuous	The average number of packets in a sub flow in the forward direction.
subflow_fwd_bytes	Continuous	The average number of bytes in a sub flow in the forward direction.
subflow_bwd_packets	Continuous	The average number of packets in a sub flow in the backward direction.
subflow_bwd_bytes	Continuous	The average number of bytes in a sub flow in the backward direction.
init_win_bytes_forward	Continuous	The total number of bytes sent in initial window in the forward direction.
init_win_bytes_backward	Continuous	The total number of bytes sent in initial window in the backward direction.
act_data_pkt_fwd	Continuous	Actual number of packets sent in the forward direction.
min_seg_size_forward	Continuous	Minimum size of the data segment sent in the forward direction.
active_mean	Continuous	The average time a flow was active before becoming idle.
active_std	Continuous	The standard deviation time a flow was active before becoming idle.
active_max	Continuous	The maximum time a flow was active before becoming idle.
active_min	Continuous	The minimum time a flow was active before becoming idle.
idle_mean	Continuous	The average time a flow was idle before becoming active.
idle_std	Continuous	The standard deviation time a flow was idle before becoming active.
idle_max	Continuous	The maximum time a flow was idle before becoming active.
idle_min	Continuous	The minimum time a flow was idle before becoming active.
label	Categorical(Target)	BENIGN: Normal legitimate network traffic | DDoS: Malicious network traffic

```r
library(performance)
library(caret)
library(ResourceSelection)
```

```r
DDoS_dataset <- read_csv("Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv") %>% clean_name
# recode label variable to 1 and 0.
DDoS_dataset <- DDoS_dataset %>%
  mutate(label = ifelse(label %in% c("DDoS", "BENIGN"),
                        dplyr::recode(label, "DDoS" = 1, "BENIGN" = 0),
                        label))

# converting several variables to factor
DDoS_dataset <- DDoS_dataset %>%
  mutate(
    label = as.factor(label),
    protocol = as.factor(protocol),
    fwd_psh_flags = as.factor(fwd_psh_flags),
    bwd_psh_flags = as.factor(bwd_psh_flags),
    fwd_urg_flags = as.factor(fwd_urg_flags),
    bwd_urg_flags = as.factor(bwd_urg_flags),
    fin_flag_count = as.factor(fin_flag_count),
    syn_flag_count = as.factor(syn_flag_count),
    rst_flag_count = as.factor(rst_flag_count),
    psh_flag_count = as.factor(psh_flag_count),
    ack_flag_count = as.factor(ack_flag_count),
    urg_flag_count = as.factor(urg_flag_count),
    cwe_flag_count = as.factor(cwe_flag_count),
    ece_flag_count = as.factor(ece_flag_count),
    flow_packets_s = as.numeric(flow_packets_s),
    flow_bytes_s = as.numeric(flow_bytes_s)
  )

DDoS_dataset <- DDoS_dataset %>%
  filter(is.finite(as.numeric(as.character(flow_packets_s))) &
         is.finite(as.numeric(as.character(flow_bytes_s))))

# --- check for null values
null_value <- sum(is.na(DDoS_dataset))
```

```r
DDoS_dataset0 <- DDoS_dataset %>%
  dplyr::select(-flow_id, -source_ip,
         -source_port, -destination_ip,
         -destination_port, -timestamp,
         -protocol, -syn_flag_count,
         -fin_flag_count, -urg_flag_count,
         -ece_flag_count, -rst_flag_count,
         -cwe_flag_count, -fwd_psh_flags,
         -bwd_psh_flags, -fwd_urg_flags,
         -bwd_urg_flags, -fwd_avg_bytes_bulk,
         -fwd_avg_packets_bulk, -fwd_avg_bulk_rate,
         -bwd_avg_bytes_bulk, -bwd_avg_bulk_rate,
         -bwd_avg_packets_bulk, -init_win_bytes_forward,
         -init_win_bytes_backward)
```

```r
# Remove rows that falls outside the 2.5-97.5 percentile range
DDoS_dataset_trimmed <- DDoS_dataset0 %>%
  mutate(across(where(is.numeric), ~ ifelse(
    . < quantile(., 0.025, na.rm = TRUE) |
    . > quantile(., 0.975, na.rm = TRUE),
    NA, .))) %>%
  drop_na()

# Using a log transformation to address right-skwness on most of the numeric values
LogScaled_Data <- DDoS_dataset_trimmed %>%
  dplyr::select(-active_std) %>%
  mutate(across(where(is.numeric), ~ ifelse(. == 0, log1p(.), log(.))))
```

```r
set.seed(000) # setting seed for reproducibility
# randomly sampling 15,000 rows to be used for the modeling
SampledData <- LogScaled_Data %>% sample_n(size = 10000)
```

```r
# --- remove unwanted variables
FinalData <-  SampledData %>%
  dplyr::select(label, psh_flag_count, ack_flag_count ,down_up_ratio, flow_packets_s,
                total_backward_packets, fwd_iat_min, flow_iat_min,
                idle_std, bwd_iat_min, fwd_iat_total, min_packet_length,
                min_seg_size_forward, idle_mean,
                total_length_of_fwd_packets,bwd_header_length,subflow_fwd_packets)
```

```r
# fitting a model without ack_flag_count, total_backward_packets, fwd_header_length_41, to
InitialModel1 <- glm(label ~ psh_flag_count + down_up_ratio + flow_packets_s +
                       fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min +
                       fwd_iat_total + idle_mean + total_backward_packets,
                           family = binomial(link = "logit"),
                           data = FinalData,
                           control = glm.control(maxit = 1000))

# step function for model 1
STEP_RESULTS1 <- stepAIC(InitialModel1, direction = 'both')
# AIC VALUE for model 1
AIC(STEP_RESULTS1)
```

```r
# fitting a model without psh_flag_count, total_backward_packets, fwd_header_length_41,min_
InitialModel2 <- glm(label ~ ack_flag_count + down_up_ratio + flow_packets_s +
                       fwd_iat_min + flow_iat_min + idle_std +
                       bwd_iat_min + fwd_iat_total + min_packet_length + idle_mean,
                           family = binomial(link = "logit"),
                           data = FinalData,
                           control = glm.control(maxit = 1000))

# step function for model 2
STEP_RESULTS2 <- stepAIC(InitialModel2, direction = 'both')
# AIC VALUE for model 2
AIC(STEP_RESULTS2)
```

```r
# Fitting normal model
InitialModel3 <- glm(label ~ down_up_ratio + idle_mean +
                            flow_packets_s  + psh_flag_count +
                            fwd_iat_min + flow_iat_min + idle_std +
                            bwd_iat_min + subflow_fwd_packets + bwd_header_length,
                      family = binomial(link = "logit"),
                      data = FinalData,
                      control = glm.control(maxit = 1000))

STEP_RESULTS3 <- stepAIC(InitialModel3, direction = 'both')
AIC(STEP_RESULTS3)
```

```r
# refitted model 3 with interaction effects
MODEL3 <- glm(label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
                      fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min +
                      subflow_fwd_packets + bwd_header_length + bwd_iat_min:down_up_ratio +
                      flow_packets_s:psh_flag_count + idle_mean:bwd_iat_min +
                      idle_mean:flow_packets_s +psh_flag_count:flow_iat_min +
                      flow_iat_min:idle_std, family = binomial(link = "logit"),
                      data = FinalData, control = glm.control(maxit = 1000))
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##                  Term  VIF   VIF 95% CI Increased SE Tolerance Tolerance 95% CI
##         down_up_ratio 3.68 [3.55, 3.80]         1.92      0.27     [0.26, 0.28]
##             idle_mean 3.48 [3.37, 3.60]         1.87      0.29     [0.28, 0.30]
##        flow_packets_s 2.60 [2.52, 2.69]         1.61      0.38     [0.37, 0.40]
##        psh_flag_count 3.68 [3.56, 3.81]         1.92      0.27     [0.26, 0.28]
##           fwd_iat_min 3.38 [3.27, 3.50]         1.84      0.30     [0.29, 0.31]
##          flow_iat_min 3.78 [3.65, 3.91]         1.94      0.26     [0.26, 0.27]
##              idle_std 1.47 [1.43, 1.51]         1.21      0.68     [0.66, 0.70]
##           bwd_iat_min 2.11 [2.05, 2.17]         1.45      0.47     [0.46, 0.49]
##   subflow_fwd_packets 3.72 [3.59, 3.84]         1.93      0.27     [0.26, 0.28]
##     bwd_header_length 3.76 [3.63, 3.89]         1.94      0.27     [0.26, 0.28]
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##                           Term  VIF   VIF 95% CI Increased SE Tolerance
##                    fwd_iat_min 3.24 [3.14, 3.35]         1.80      0.31
##                   flow_iat_min 4.71 [4.55, 4.88]         2.17      0.21
##                       idle_std 2.72 [2.63, 2.80]         1.65      0.37
##                    bwd_iat_min 3.65 [3.53, 3.78]         1.91      0.27
##            subflow_fwd_packets 4.22 [4.08, 4.37]         2.05      0.24
##              bwd_header_length 4.01 [3.87, 4.15]         2.00      0.25
##      down_up_ratio:bwd_iat_min 2.70 [2.61, 2.79]         1.64      0.37
##          idle_mean:bwd_iat_min 3.26 [3.16, 3.37]         1.81      0.31
##       idle_mean:flow_packets_s 3.96 [3.83, 4.10]         1.99      0.25
##  psh_flag_count:flow_iat_min 4.29 [4.14, 4.44]         2.07      0.23
##          flow_iat_min:idle_std 1.29 [1.26, 1.32]         1.14      0.77
##  Tolerance 95% CI
##      [0.30, 0.32]
##      [0.20, 0.22]
##      [0.36, 0.38]
##      [0.26, 0.28]
##      [0.23, 0.25]
##      [0.24, 0.26]
##      [0.36, 0.38]
##      [0.30, 0.32]
##      [0.24, 0.26]
##      [0.23, 0.24]
##      [0.76, 0.79]
##
## Moderate Correlation
##
##                            Term  VIF   VIF 95% CI Increased SE Tolerance
##                   down_up_ratio 5.85 [5.65, 6.07]         2.42      0.17
##                       idle_mean 5.16 [4.98, 5.34]         2.27      0.19
##                  flow_packets_s 6.27 [6.05, 6.51]         2.50      0.16
##                  psh_flag_count 7.91 [7.63, 8.21]         2.81      0.13
##  flow_packets_s:psh_flag_count 6.83 [6.59, 7.08]         2.61      0.15
##  Tolerance 95% CI
##      [0.16, 0.18]
##      [0.19, 0.20]
##      [0.15, 0.17]
##      [0.12, 0.13]
##      [0.14, 0.15]
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.62067 -0.01945  0.16140  0.08021  0.35259  2.97580
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  as.numeric(as.character(SampledData$label)), fitted(MODEL3)
## X-squared = 12.554, df = 8, p-value = 0.1281
```

```
## Analysis of Deviance Table
##
## Model 1: label ~ 1
## Model 2: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_header_length + bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      9999    11786.6
## 2      9983     4205.9 16   7580.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Analysis of Deviance Table
##
```

```
## Model 1: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + bwd_header_length +
##     bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
## Model 2: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_header_length + bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      9984     4456.8
## 2      9983     4205.9  1   250.92 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

## Analysis of Deviance Table
##
## Model 1: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
## Model 2: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_header_length + bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      9984     5332.1
## 2      9983     4205.9  1   1126.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

## Analysis of Deviance Table
##
## Model 1: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_header_length
## Model 2: label ~ down_up_ratio + idle_mean + flow_packets_s + psh_flag_count +
##     fwd_iat_min + flow_iat_min + idle_std + bwd_iat_min + subflow_fwd_packets +
##     bwd_header_length + bwd_iat_min:down_up_ratio + flow_packets_s:psh_flag_count +
##     idle_mean:bwd_iat_min + idle_mean:flow_packets_s + psh_flag_count:flow_iat_min +
##     flow_iat_min:idle_std
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      9989     5437.5
## 2      9983     4205.9  6   1231.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2251  139
##          1  511 7099
##
##                Accuracy : 0.935
##                  95% CI : (0.93, 0.9398)
##     No Information Rate : 0.7238
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8304
##
##  Mcnemar’s Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9808
##             Specificity : 0.8150
##          Pos Pred Value : 0.9329
##          Neg Pred Value : 0.9418
##              Prevalence : 0.7238
##          Detection Rate : 0.7099
##    Detection Prevalence : 0.7610
##       Balanced Accuracy : 0.8979
##
##        ’Positive’ Class : 1
##
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

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

```
##                                      2.5 %      97.5 %
## (Intercept)                     8.56295014  9.96213454
## down_up_ratio                  -2.95355012 -2.24555236
## idle_mean                      -0.10925093 -0.05883100
## flow_packets_s                 -1.16406007 -0.96931508
## psh_flag_count1                 0.61611548  1.53926111
## fwd_iat_min                     0.22327353  0.34966971
## flow_iat_min                   -0.74886243 -0.54086540
## idle_std                       -0.02810631  0.03329417
## bwd_iat_min                    -0.05875836  0.11261005
## subflow_fwd_packets            -2.37696133 -1.83414008
## bwd_header_length              -1.55712405 -1.33604287
## down_up_ratio:bwd_iat_min       0.36684907  0.57118082
## flow_packets_s:psh_flag_count1  1.18078149  1.42798108
## idle_mean:bwd_iat_min           0.02695192  0.04199504
## idle_mean:flow_packets_s       -0.09130649 -0.05742057
## psh_flag_count1:flow_iat_min    0.35557301  0.62019509
## flow_iat_min:idle_std           0.02377912  0.09498267
```