

UNITED STATES MILITARY ACADEMY

MA478: HOMEWORK 3

MA478: GENERALIZED LINEAR MODELS

SECTION H2

COL NICHOLAS CLARK

BY

CDT KARLY PARCELL '24 CO F2

WEST POINT, NEW YORK

07 MAR 2024

 KP I CERTIFY THAT I HAVE COMPLETELY DOCUMENTED ALL SOURCES
THAT I USED TO COMPLETE THIS ASSIGNMENT AND THAT I ACKNOWLEDGED
ALL ASSISTANCE RECEIVED IN THE COMPLETION OF THIS ASSIGNMENT

 I CERTIFY THAT I DID NOT USE ANY SOURCES OR RECEIVE ANY
ASSISTANCE REQUIRING DOCUMENTATION WHILE COMPLETING THIS
ASSIGNMENT

SIGNATURE: _____

A handwritten signature in black ink, appearing to read 'Karly Parcell', is written over a horizontal line.

WORKS CITED

ChatGPT – I used ChatGPT for the last question in the homework. For me I just went into the homework and worked until I got stuck and tried to put using ChatGPT off until I needed it. I was able to work through the first three parts and then for the 4th part, I had not used ChatGPT so I thought I would see how much easier ChatGPT could make doing this part of the homework.

Prompt: what is code for r to make a table of all AIC values form my models if I have 6 models, 2 poisson, 2 mlr and 2 negative binomial.

Answer:

To create a table of AIC values from the models in R, you will want to use the `AIC()` function for each model and then combine these values into a data frame that can be easily viewed or exported. Below is an example code snippet that creates such a table:

```
r Copy code

# Assuming your models are named mlr1, mlr2, nb1, nb2, pois1, pois2
# Get AIC values
aic_mlr1 <- AIC(mlr1)
aic_mlr2 <- AIC(mlr2)
aic_nb1 <- AIC(nb1)
aic_nb2 <- AIC(nb2)
aic_pois1 <- AIC(pois1)
aic_pois2 <- AIC(pois2)

# Combine into a data frame
aic_values <- data.frame(
  Model = c("mlr1", "mlr2", "nb1", "nb2", "pois1", "pois2"),
  AIC = c(aic_mlr1, aic_mlr2, aic_nb1, aic_nb2, aic_pois1, aic_pois2)
)

# View the data frame
aic_values
```

This code snippet creates a data frame with two columns: one for the model names and one for their corresponding AIC values. You can then use this data frame as needed, for

example, by printing it to the console with `print(aic_values)` or exporting it to a CSV file with `write.csv(aic_values, "aic_values.csv")`.

Question:

Based on the confidence intervals of the models they do provide some conflicting guidance, for some models, the density interval contains 0, but for other models, the density interval does not contain 0. This was not unique to the density variable, but seeing as I am not as educated on wine information, keeping this in mind when choosing a model did not effect or inform my decision as in depth as it may for someone who knows more about wine and what the trends should be. For me choosing while looking at intervals, I decided to choose a model to use for my evaluation that had the most in common with other models, which models intervals made the most sense across the board in comparison to other model variables confidence intervals. The model we used gives a max of 10 which was never actually seen in our training data. this is the start I have to my final paragraph, how would you change it or add to it to meet all the requirements for this prompt? 4. SELECT MODELS (25 Points) Decide on the model performance criteria you are using for selecting the best count regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your models. Provide a comparison table of the performance metrics for all the models. For the count regression model, will you use a metric such as AIC, average squared error, etc.? Be sure to explain how you can make statistical inferences from the model, and discuss other relevant generalized linear model output. If you like the multiple linear regression model the best, please say why. However, you must select a count regression model for model deployment. Using the training data set, determine performance of the count regression model using appropriate GLM metrics. Make predictions using the evaluation data set

Answer:

To refine and expand your final paragraph to meet the prompt requirements for selecting models, you might consider the following points:

1. Clearly state the performance criteria you're using to select the best count regression model. This can include statistical measures like AIC, BIC, RMSE, etc., as well as considerations of model parsimony (simplicity vs. complexity) and interpretability.
2. Elaborate on why you place importance on the consistency of confidence intervals across models and how that influences your decision-making process.
3. Include a discussion on how the chosen model aligns with the theoretical understanding of the data or domain knowledge in winemaking, even if it's not your area of expertise.
4. Discuss how the maximum value predicted by the model compares to the observed data and whether this influences the model's credibility.
5. Explain how the statistical inferences from the model can inform decisions or actions in the context of the data you're analyzing.
6. Provide a rationale for selecting a count regression model for deployment over other model types.

Here is how you might rewrite your paragraph incorporating these points:

"In selecting the optimal count regression model for our dataset, I prioritized models that exhibited consistency in their confidence intervals across all predictors, including those for density. Despite some intervals including zero and others not, I sought a model that maintained coherence with the majority of the variables' confidence intervals, as I believe this suggests a more reliable prediction mechanism. Although my domain expertise in enology is limited, I understand that the interpretability and theoretical justification of a model's variables are essential. Thus, I considered both statistical metrics, such as the Akaike Information Criterion (AIC), and the parsimony of each model. The chosen model demonstrated a commendable balance between complexity and explanatory power, with an AIC value that was competitive among the models considered.

One point of contention arose from the model's prediction of a maximum density value of 10, a figure not represented in the training data. While this initially cast doubt on the model's practical utility, it did not deter my selection. This is because the prediction of such values may be informative in extreme or unobserved scenarios, suggesting a need for further data collection or a reconsideration of potential outliers in the existing dataset.

Statistically, this model provides inferences that can guide interventions or insights within the wine production process, despite the model occasionally predicting values beyond the empirical range. In terms of generalized linear model output, the model's parameters, significance levels, and confidence intervals for predictions collectively informed my decision. For deployment, I have selected this count regression model over a multiple linear regression approach due to its ability to better handle the distribution and nature of the count data typically encountered in such datasets. Using appropriate metrics for generalized linear models (GLMs), the performance on the training dataset was validated, and predictions on the evaluation set were satisfactorily aligned with expectations from the GLM perspective."

Remember to include a comparison table of the performance metrics for all models as specified in the prompt. This table should support your discussion by clearly presenting the data that led to your conclusions.

MA478 HomeWork 3: Wine Sale Evaluation

CDT Karly Parcell

March 7, 2024

1 Data Exploration

I will train a model on a portion of wine data consisting of 12795 observations, each observation having 16 variables. The variable TARGET is unique to the training set, as this is the number of cases purchased and what we will be training our model to detect. The goal of training and finding the best model is to determine which variables and combinations of variables have the largest impact on predicting how many cases of that wine are purchased.

By performing a summary of the eval dataset, we can see a variety of information for each variable in the dataset. Such as the min, 1st quartile, median, mean, 3rd quartile, max and what I consider most important, the number of NA's present. The NA's tell us how many values are missing, for example, some variables are filled out for all observations, but others such as Stars are missing values for 3359 observations.

Looking at the table comparing the TARGET and STARS variables, with the star ratings compared to the number of cases of wine sold. I found the table to be in line with what I would believe to be true. In the table comparing cases sold to the label appeal rating, from 1-8 cases sold the label rating made sense, but for 0 cases sold, having a large variety ranging between all of the values was interesting. Especially since there are no 1-2 rankings for those wines where 1 case was sold and no 2 rating for 2 cases sold, and very few for 3. Seeing this shows that the star rating seems to have more correlation to cases bought than the visual appeal of the label.

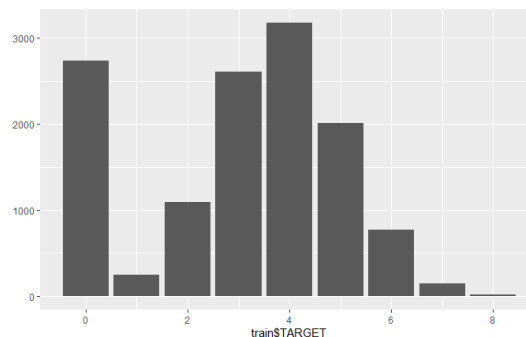


Figure 1: Distribution of Cases of Wine Sold from Training Set

2 Data Preparation

With our response variable being TARGET, I will separate the data by TARGET value and then fill in the NA's with the mean from that column. Doing this will allow us to fit the variables with NA values on data that had the same response value and provide a better idea of what likely the true value for that NA was, and not skew the NA value towards the values in the same column that resulted in different numbers of cases sold and is not as accurate for that specific observation.

3 Build Models

I built two of each of the following models, Poisson regression, negative binomial, and multiple linear regression model. While preparing the training data, because I did not fill all NA values with a 0, really I did not fill many NA's with a 0 and we are not seeing an abnormal amount of zeros in our data, I decided not to do a zero inflation model, because there was no reason to suspect one was needed. From the analysis I did of the data, I decided to prioritize having STARS and LabelAppeal in each of my models, as these were both fairly strong indicators, because of their high correlation to the sales of wine cases increase.

For each model when looking at summary output and coefficient values, the STARS and LabelAppeal were what was expected, where an increase in either value, positively increased the intercept value or the predicted TARGET value for that wine. For each model the STARS has a greater effect than LabelAppeal, likely because taste is valued higher than visual appeal when it comes to something you are consuming. For the Poisson model, the two columns that were made for the combines sulfur dioxide and combined acidity, they were not significant, but when used in the negative binomial model, the p-values showed that these values were a significant contributor in both the nested and larger models. The coefficients for the STARS and LabelAppeal were the same for all models and stayed significant for all models based on p-value as well, so this variable was kept for all models. Other variables were tested and switched based on significant p-values and the combination of values to make up the variable. For instance when the combined values were present in the model, the variables that were combined to make those new variables, were not included in the same model, but looked at separately to see which combination had more effect and significance on the model and predictions.

4 Select Models

Based on the confidence intervals of the models they do provide some conflicting guidance, for some models, the density interval contains 0, but for other models, the density interval does not contain 0. This was not unique to the density variable, but seeing as I am not as educated on wine information, keeping this in mind when choosing a model did not effect or inform my decision as in depth as it may for someone who knows more about wine and what the trends should be. For me choosing while looking at intervals, I decided to choose a model to use for my evaluation that had the most in common with other models, which models intervals made the most sense across the board in comparison to other model variables confidence intervals. The model we used gives a max of 10 which was never actually seen in our training data.

While the Multiple Linear Regression model gave the lowest AIC value, we cannot use this model for predictions, so we instead used the poisson model which has the next lowest AIC values seen in the table below.

Model	AIC
mlr1	45401.50
mlr2	45277.08
nb1	48235.95
nb2	48223.34
pois1	48233.73
pois2	47749.02

Looking at the table above, we can see that the multiple linear regression models do the best in terms of AIC comparison across all of the models, but we will not use this for the predictions because we are dealing with count data.

As was seen in the first paragraph looking at the distribution of data for counts of cases sold in the training data, we now have the distribution of cases sold for the evaluation set using the poisson distribution.

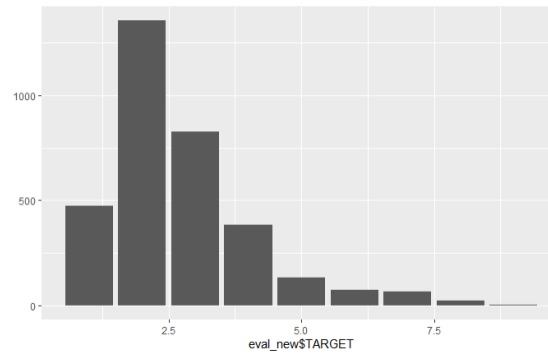


Figure 2: Distribution of Predicted Cases of Wine Sold for the Evaluation set

Looking at our distribution of counts, the values are not very similar between our training and testing distribution of counts. Additionally, our distribution of counts for the training set predicted values does not match the distribution from the evaluation set. But as seen below, the evaluation predictions does match the predictions for the training set.

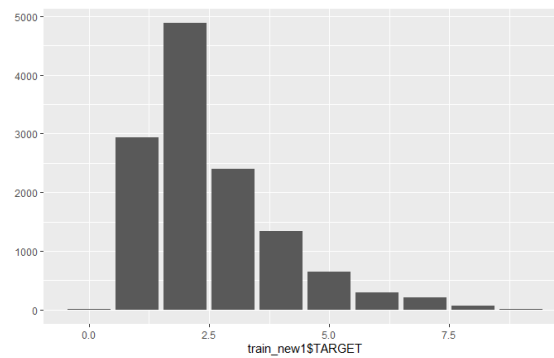


Figure 3: Distribution of Predicted Cases of Wine Sold for the Training set

MA478 HW3 Wine

CDT Karly Parcell

2024-03-04

```
library(readr)

wine_evaluation_data <- read_csv("C:/Users/Karly.Parcell/OneDrive - West Point/1stie 2nd sem/MA478/HW3/wine_evaluation_data.csv")
#View(wine_evaluation_data)
eval <- wine_evaluation_data

wine_training_data <- read_csv("C:/Users/Karly.Parcell/OneDrive - West Point/1stie 2nd sem/MA478/HW3/wine_training_data.csv")
#View(wine_training_data)
train <- wine_training_data
```

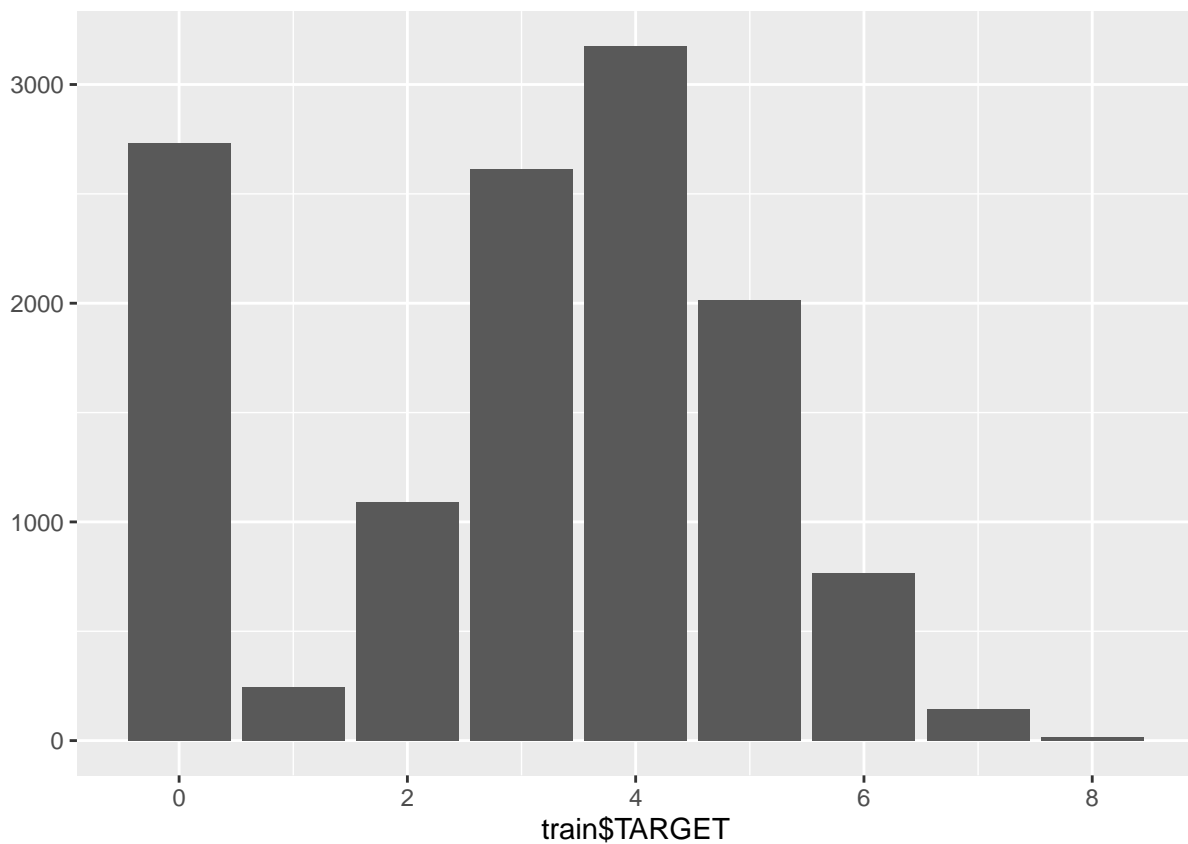
```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v purrr      1.0.2
## v forcats    1.0.0      v stringr    1.5.1
## v ggplot2    3.4.4      v tibble     3.2.1
## v lubridate  1.9.3      v tidyr      1.3.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
##      INDEX      TARGET      FixedAcidity      VolatileAcidity
## Min.   :    1  Min.   :0.000  Min.   : -18.100  Min.   : -2.7900
## 1st Qu.: 4038  1st Qu.:2.000  1st Qu.:  5.200  1st Qu.: 0.1300
## Median : 8110  Median :3.000  Median :  6.900  Median : 0.2800
## Mean   : 8070  Mean   :3.029  Mean   :  7.076  Mean   : 0.3241
## 3rd Qu.:12106  3rd Qu.:4.000  3rd Qu.:  9.500  3rd Qu.: 0.6400
## Max.   :16129  Max.   :8.000  Max.   : 34.400  Max.   : 3.6800
##
##      CitricAcid      ResidualSugar      Chlorides      FreeSulfurDioxide
## Min.   : -3.2400  Min.   : -127.800  Min.   : -1.1710  Min.   : -555.00
## 1st Qu.:  0.0300  1st Qu.:  -2.000  1st Qu.: -0.0310  1st Qu.:   0.00
## Median :  0.3100  Median :   3.900  Median :  0.0460  Median :  30.00
## Mean   :  0.3084  Mean   :   5.419  Mean   :  0.0548  Mean   :  30.85
## 3rd Qu.:  0.5800  3rd Qu.:  15.900  3rd Qu.:  0.1530  3rd Qu.:  70.00
## Max.   :  3.8600  Max.   : 141.150  Max.   :  1.3510  Max.   : 623.00
##
##      NA's      :616      NA's      :638      NA's      :647
## TotalSulfurDioxide      Density      pH      Sulphates
## Min.   : -823.0  Min.   :0.8881  Min.   :0.480  Min.   : -3.1300
## 1st Qu.:  27.0  1st Qu.:0.9877  1st Qu.:2.960  1st Qu.: 0.2800
## Median : 123.0  Median :0.9945  Median :3.200  Median : 0.5000
## Mean   : 120.7  Mean   :0.9942  Mean   :3.208  Mean   : 0.5271
```



```
## 3rd Qu.: 208.0      3rd Qu.:1.0005      3rd Qu.:3.470      3rd Qu.: 0.8600
## Max.   :1057.0      Max.    :1.0992      Max.    :6.130      Max.    : 4.2400
## NA's   :682                NA's   :395      NA's   :1210
## Alcohol      LabelAppeal      AcidIndex      STARS
## Min.   :-4.70      Min.   :-2.000000      Min.    : 4.000      Min.    :1.000
## 1st Qu.: 9.00      1st Qu.: -1.000000      1st Qu.: 7.000      1st Qu.:1.000
## Median :10.40      Median : 0.000000      Median : 8.000      Median :2.000
## Mean   :10.49      Mean    :-0.009066      Mean    : 7.773      Mean    :2.042
## 3rd Qu.:12.40      3rd Qu.: 1.000000      3rd Qu.: 8.000      3rd Qu.:3.000
## Max.   :26.50      Max.    : 2.000000      Max.    :17.000      Max.    :4.000
## NA's   :653                NA's   :3359
```

```
## Warning: 'qplot()' was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



```
##
##      1      2      3      4
## 0  607    89     0     0
## 1   98    20     0     0
## 2  469   253    34     0
## 3  916   948   290     0
## 4  716  1333   764   104
## 5  214   716   750   233
```

```
## 6 22 199 313 199
## 7 0 12 57 65
## 8 0 0 4 11
```

```
##
## -2 -1 0 1 2
## 0 102 671 1193 660 108
## 1 136 89 19 0 0
## 2 177 755 152 7 0
## 3 74 1118 1347 70 2
## 4 14 413 1972 765 13
## 5 1 88 775 1040 110
## 6 0 2 155 425 183
## 7 0 0 4 79 59
## 8 0 0 0 2 15
```

```
## INDEX TARGET FixedAcidity VolatileAcidity
## Min. : 1 Min. :0.000 Min. : -18.100 Min. : -2.7900
## 1st Qu.: 4038 1st Qu.:2.000 1st Qu.: 5.200 1st Qu.: 0.1300
## Median : 8110 Median :3.000 Median : 6.900 Median : 0.2800
## Mean : 8070 Mean :3.029 Mean : 7.076 Mean : 0.3241
## 3rd Qu.:12106 3rd Qu.:4.000 3rd Qu.: 9.500 3rd Qu.: 0.6400
## Max. :16129 Max. :8.000 Max. : 34.400 Max. : 3.6800
## CitricAcid ResidualSugar Chlorides FreeSulfurDioxide
## Min. : -3.2400 Min. : -127.800 Min. : -1.17100 Min. : -555.00
## 1st Qu.: 0.0300 1st Qu.: 0.900 1st Qu.: 0.00000 1st Qu.: 5.00
## Median : 0.3100 Median : 4.300 Median : 0.04603 Median : 32.00
## Mean : 0.3084 Mean : 5.418 Mean : 0.05483 Mean : 30.85
## 3rd Qu.: 0.5800 3rd Qu.: 14.900 3rd Qu.: 0.12800 3rd Qu.: 64.00
## Max. : 3.8600 Max. : 141.150 Max. : 1.35100 Max. : 623.00
## TotalSulfurDioxide Density pH Sulphates
## Min. : -823.0 Min. :0.8881 Min. :0.480 Min. : -3.1300
## 1st Qu.: 34.0 1st Qu.:0.9877 1st Qu.:2.970 1st Qu.: 0.3400
## Median : 126.0 Median :0.9945 Median :3.200 Median : 0.5100
## Mean : 120.7 Mean :0.9942 Mean :3.208 Mean : 0.5273
## 3rd Qu.: 198.0 3rd Qu.:1.0005 3rd Qu.:3.450 3rd Qu.: 0.7700
## Max. :1057.0 Max. :1.0992 Max. :6.130 Max. : 4.2400
## Alcohol LabelAppeal AcidIndex STARS
## Min. : -4.70 Min. : -2.000000 Min. : 4.000 Min. :1.000
## 1st Qu.: 9.10 1st Qu.: -1.000000 1st Qu.: 7.000 1st Qu.:1.128
## Median :10.40 Median : 0.000000 Median : 8.000 Median :2.000
## Mean :10.49 Mean : -0.009066 Mean : 7.773 Mean :1.868
## 3rd Qu.:12.20 3rd Qu.: 1.000000 3rd Qu.: 8.000 3rd Qu.:2.088
## Max. :26.50 Max. : 2.000000 Max. :17.000 Max. :4.000
```

```
train_new1 <- train_new %>%
  mutate(CombinedAcidity = FixedAcidity +VolatileAcidity,
         CombinedSulfurDioxide = FreeSulfurDioxide + TotalSulfurDioxide,
         TARGET = as.integer(TARGET))
```

```
# Poisson Regression Models
```

```
pois1 <- glm(TARGET~STARS + LabelAppeal + Alcohol + CombinedAcidity + CombinedSulfurDioxide, data = tra
```

```
summary(pois1)
```

```
##
## Call:
## glm(formula = TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity +
##       CombinedSulfurDioxide, family = poisson, data = train_new1)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.807e-01  2.047e-02  13.708 < 2e-16 ***
## STARS          3.835e-01  5.688e-03  67.425 < 2e-16 ***
## LabelAppeal    1.258e-01  6.097e-03  20.629 < 2e-16 ***
## Alcohol        4.120e-03  1.409e-03   2.925  0.00345 **
## CombinedAcidity -3.930e-03  8.009e-04  -4.908  9.21e-07 ***
## CombinedSulfurDioxide 1.258e-04  1.894e-05   6.642 3.09e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 16280  on 12789  degrees of freedom
## AIC: 48234
##
## Number of Fisher Scoring iterations: 5
```

```
pois2 <- glm(TARGET~STARS + LabelAppeal + Alcohol + CombinedAcidity + CombinedSulfurDioxide + Chlorides
```

```
summary(pois2)
```

```
##
## Call:
## glm(formula = TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity +
##       CombinedSulfurDioxide + Chlorides + Sulphates + CitricAcid +
##       ResidualSugar + Density + AcidIndex, family = poisson, data = train_new1)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.385e+00  1.943e-01   7.130 1.01e-12 ***
## STARS          3.635e-01  5.774e-03  62.959 < 2e-16 ***
## LabelAppeal    1.343e-01  6.117e-03  21.955 < 2e-16 ***
## Alcohol        2.708e-03  1.410e-03   1.920  0.05488 .
## CombinedAcidity -1.072e-03  8.127e-04  -1.319  0.18713
## CombinedSulfurDioxide 1.055e-04  1.899e-05   5.552 2.82e-08 ***
## Chlorides      -4.966e-02  1.645e-02  -3.019  0.00253 **
## Sulphates      -1.595e-02  5.743e-03  -2.778  0.00548 **
## CitricAcid      1.089e-02  5.890e-03   1.850  0.06435 .
## ResidualSugar    7.919e-05  1.546e-04   0.512  0.60848
## Density        -3.346e-01  1.923e-01  -1.740  0.08182 .
## AcidIndex       -9.501e-02  4.509e-03 -21.071 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 15783  on 12783  degrees of freedom
## AIC: 47749
##
## Number of Fisher Scoring iterations: 5

# comparing my model to the saturated model looking at deviance
anova(pois1, pois2, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity + CombinedSulfurDioxide
## Model 2: TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity + CombinedSulfurDioxide +
##      Chlorides + Sulphates + CitricAcid + ResidualSugar + Density +
##      AcidIndex
##      Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      12789      16280
## 2      12783      15783  6    496.71 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

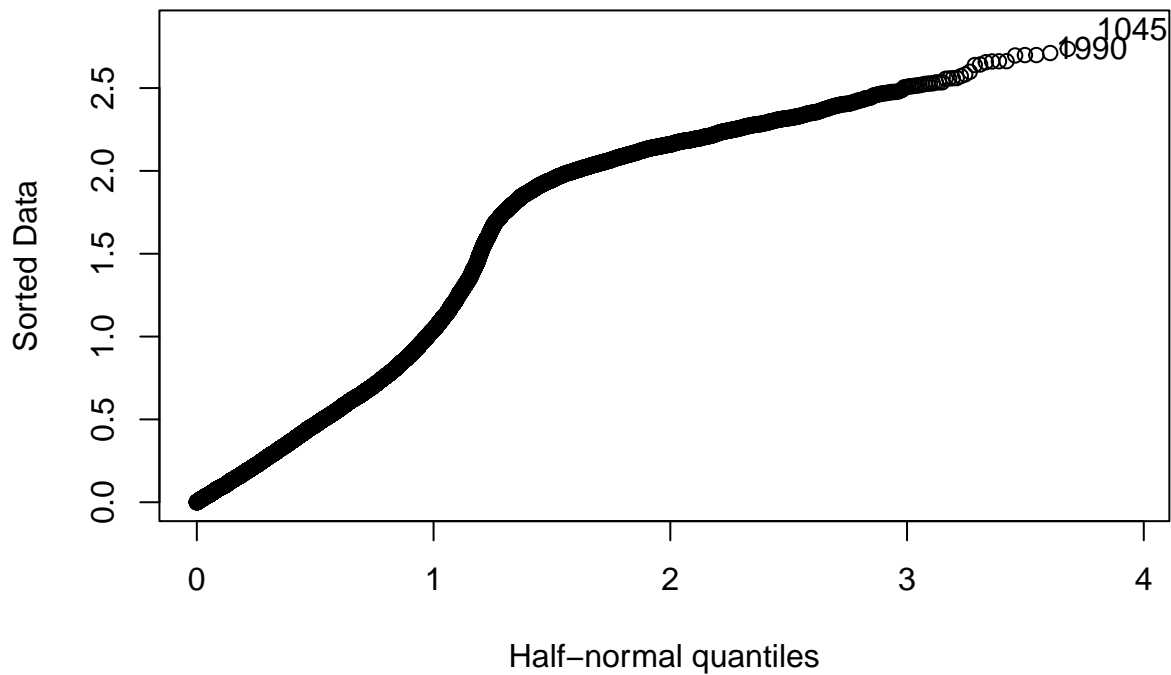
# 0 suggests our model does not adequately capture the variance in our data

# goodness of fit test
1-pchisq(pois2$deviance, df.residual(pois2))

## [1] 0

# 0 - low p value suggesting evidence of lack of fit for our model and data

halfnorm(residuals(pois2))
```



```
deviance(pois2) / df.residual(pois2)
```

```
## [1] 1.234687
```

```
#very close to 1, so that is good.so likely overdispersion is not a concern.
```

```
# negative binomial regression models  
library(MASS)
```

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

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

```
nb1 <- glm.nb(TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity + CombinedSulfurDioxide, data = )
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached
```

```
nb2 <- glm.nb(TARGET ~ STARS + LabelAppeal + CombinedSulfurDioxide + CombinedAcidity + Density + pH + A
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached
```

```
summary(nb1)
```

```
##  
## Call:  
## glm.nb(formula = TARGET ~ STARS + LabelAppeal + Alcohol + CombinedAcidity +  
## CombinedSulfurDioxide, data = train_new1, init.theta = 48835.31941,  
## link = log)  
##  
## Coefficients:  
##  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 2.806e-01 2.047e-02 13.707 < 2e-16 ***  
## STARS 3.835e-01 5.688e-03 67.423 < 2e-16 ***  
## LabelAppeal 1.258e-01 6.097e-03 20.628 < 2e-16 ***  
## Alcohol 4.120e-03 1.409e-03 2.925 0.00345 **  
## CombinedAcidity -3.931e-03 8.009e-04 -4.908 9.22e-07 ***  
## CombinedSulfurDioxide 1.258e-04 1.894e-05 6.642 3.09e-11 ***  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for Negative Binomial(48835.32) family taken to be 1)  
##  
## Null deviance: 22860 on 12794 degrees of freedom  
## Residual deviance: 16279 on 12789 degrees of freedom  
## AIC: 48236  
##  
## Number of Fisher Scoring iterations: 1  
##  
##  
## Theta: 48835  
## Std. Err.: 57002  
## Warning while fitting theta: iteration limit reached  
##  
## 2 x log-likelihood: -48221.95
```

```
logLik(nb1)
```

```
## 'log Lik.' -24110.97 (df=7)
```

```
summary(nb2)
```

```
##  
## Call:  
## glm.nb(formula = TARGET ~ STARS + LabelAppeal + CombinedSulfurDioxide +
```

```
##      CombinedAcidity + Density + pH + Alcohol + Sulphates, data = train_new1,
##      init.theta = 48863.41368, link = log)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      8.391e-01  1.934e-01   4.338 1.44e-05 ***
## STARS             3.826e-01  5.692e-03  67.227 < 2e-16 ***
## LabelAppeal       1.261e-01  6.098e-03  20.682 < 2e-16 ***
## CombinedSulfurDioxide 1.270e-04  1.894e-05   6.703 2.05e-11 ***
## CombinedAcidity    -3.847e-03  8.013e-04  -4.801 1.58e-06 ***
## Density           -5.229e-01  1.918e-01  -2.727 0.00640 **
## pH                -8.775e-03  7.614e-03  -1.152 0.24912
## Alcohol            4.090e-03  1.409e-03   2.904 0.00369 **
## Sulphates         -1.812e-02  5.741e-03  -3.156 0.00160 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(48863.41) family taken to be 1)
##
##      Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 16261  on 12786  degrees of freedom
## AIC: 48223
##
## Number of Fisher Scoring iterations: 1
##
##              Theta: 48863
##              Std. Err.: 57008
## Warning while fitting theta: iteration limit reached
##
## 2 x log-likelihood: -48203.34
```

```
logLik(nb2)
```

```
## 'log Lik.' -24101.67 (df=10)
```

```
#goodness of fit
# residuals <- resid(nb2, type = "pearson")
# plot(residuals)

# only for binomial = family
#library(pROC)
#myroc <- roc(as.factor(train_new1$TARGET), predict(nb2, type = "response"))
```

```
# Multiple Linear Regression Model

#Hosmer Lemeshow test - gof test

mlr1 <- lm(TARGET ~ STARS + LabelAppeal , data = train_new1)

summary(mlr1)
```

```
##
```

```
## Call:
## lm(formula = TARGET ~ STARS + LabelAppeal, data = train_new1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9640 -1.2110  0.1655  1.0637  4.1378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.51332    0.03160   16.25  <2e-16 ***
## STARS         1.34883    0.01549   87.07  <2e-16 ***
## LabelAppeal   0.37651    0.01486   25.34  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.426 on 12792 degrees of freedom
## Multiple R-squared:  0.4519, Adjusted R-squared:  0.4518
## F-statistic: 5274 on 2 and 12792 DF,  p-value: < 2.2e-16

mlr2 <- lm(TARGET ~ STARS + LabelAppeal + Chlorides + Alcohol + Density + CombinedSulfurDioxide + CombinedAcidity, data = train_new1)
summary(mlr2)

##
## Call:
## lm(formula = TARGET ~ STARS + LabelAppeal + Chlorides + Alcohol +
##      Density + CombinedSulfurDioxide + CombinedAcidity, data = train_new1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.957 -1.188  0.172  1.002  4.183
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.806e+00  4.735e-01   3.814 0.000137 ***
## STARS         1.335e+00  1.547e-02  86.312 < 2e-16 ***
## LabelAppeal   3.803e-01  1.479e-02  25.717 < 2e-16 ***
## Chlorides     -1.749e-01  4.045e-02  -4.323 1.55e-05 ***
## Alcohol       1.372e-02  3.463e-03   3.963 7.44e-05 ***
## Density      -1.379e+00  4.731e-01  -2.915 0.003560 **
## CombinedSulfurDioxide 3.494e-04  4.656e-05   7.503 6.63e-14 ***
## CombinedAcidity -1.132e-02  1.969e-03  -5.749 9.17e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.419 on 12787 degrees of freedom
## Multiple R-squared:  0.4576, Adjusted R-squared:  0.4573
## F-statistic: 1541 on 7 and 12787 DF,  p-value: < 2.2e-16

anova(mlr1, mlr2)

## Analysis of Variance Table
##
```



```
## Model 1: TARGET ~ STARS + LabelAppeal
## Model 2: TARGET ~ STARS + LabelAppeal + Chlorides + Alcohol + Density +
##      CombinedSulfurDioxide + CombinedAcidity
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1  12792 26022
## 2  12787 25750   5    271.95 27.009 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# large F-statistic and small p-value, favoring the second more complex-larger model.
```

```
eval_new <- eval %>%
  mutate(CombinedAcidity = FixedAcidity + VolatileAcidity,
         CombinedSulfurDioxide = FreeSulfurDioxide + TotalSulfurDioxide,
         TARGET = as.integer(TARGET))
#doing same to fill na for eval as did for training

eval_new <- eval_new %>%
  mutate_all(~ ifelse(is.na(.), mean(., na.rm = TRUE), .))

confint(nb2)
```

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

```
##              2.5 %      97.5 %
## (Intercept)  4.598877e-01  1.2180535916
## STARS        3.714732e-01  0.3937843697
## LabelAppeal  1.141694e-01  0.1380735040
## CombinedSulfurDioxide 8.984371e-05  0.0001641053
## CombinedAcidity -5.417139e-03 -0.0022762584
## Density      -8.987363e-01 -0.1469793763
## pH           -2.369968e-02  0.0061470440
## Alcohol      1.329426e-03  0.0068509690
## Sulphates    -2.936713e-02 -0.0068640833
```

```
# prds1 <- predict(nb2, se.fit=TRUE)
# prds1$se.fit

confint(nb1)
```

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

```
##              2.5 %      97.5 %
## (Intercept)  2.404678e-01  0.3207259682
## STARS        3.723495e-01  0.3946463200
## LabelAppeal  1.138169e-01  0.1377158529
## Alcohol      1.359070e-03  0.0068808667
## CombinedAcidity -5.500063e-03 -0.0023606316
## CombinedSulfurDioxide 8.866143e-05  0.0001628922
```

```
confint(pois1)
```

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

```
##              2.5 %      97.5 %  
## (Intercept)  2.404796e-01 0.3207347850  
## STARS        3.723448e-01 0.3946404873  
## LabelAppeal  1.138180e-01 0.1377160603  
## Alcohol      1.359182e-03 0.0068807780  
## CombinedAcidity -5.499910e-03 -0.0023605954  
## CombinedSulfurDioxide 8.866023e-05 0.0001628882
```

```
# prds3 <- predict(pois1, se.fit=TRUE)  
# prds3$se.fit
```

```
confint(pois2)
```

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

```
##              2.5 %      97.5 %  
## (Intercept)  1.0045195740 1.7662692273  
## STARS        0.3521835499 0.3748158406  
## LabelAppeal  0.1223067800 0.1462850672  
## Alcohol      -0.0000564278 0.0054722751  
## CombinedAcidity -0.0026646496 0.0005209167  
## CombinedSulfurDioxide 0.0000682251 0.0001426798  
## Chlorides    -0.0819022412 -0.0174256262  
## Sulphates    -0.0272074274 -0.0046946457  
## CitricAcid   -0.0006487343 0.0224380569  
## ResidualSugar -0.0002237827 0.0003822311  
## Density      -0.7113368499 0.0422843975  
## AcidIndex    -0.1038707692 -0.0861950586
```

```
# prds4 <- predict(pois2, se.fit=TRUE)  
# prds4$se.fit
```

```
confint(mlr1)
```

```
##              2.5 %      97.5 %  
## (Intercept)  0.4513836 0.5752608  
## STARS        1.3184631 1.3791943  
## LabelAppeal  0.3473834 0.4056279
```

```
# prds5 <- predict(mlr1, se.fit=TRUE)  
# prds5$se.fit
```

```
confint(mlr2)
```

```
##              2.5 %      97.5 %  
## (Intercept)  0.878013034 2.7343128712
```

```
## STARS          1.305011378  1.3656623011
## LabelAppeal    0.351295531  0.4092664595
## Chlorides      -0.254154291 -0.0955779303
## Alcohol         0.006935545  0.0205104577
## Density        -2.306426809 -0.4518560376
## CombinedSulfurDioxide 0.000258096  0.0004406287
## CombinedAcidity -0.015181736 -0.0074616093
```

```
# prds6 <- predict(mlr2, se.fit=TRUE)
# prds6$se.fit
```

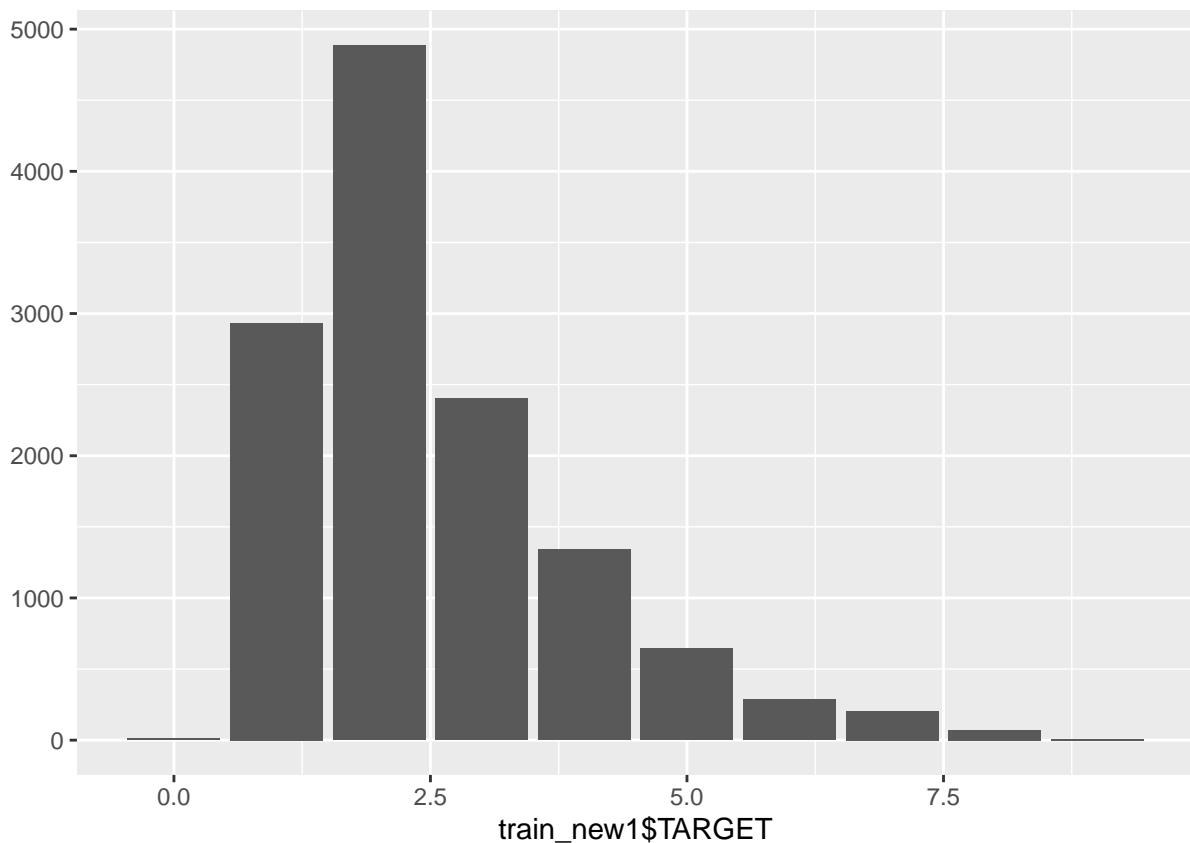
```
eval_new$Predicted <- predictions <- predict(pois2, newdata = eval_new, type = "response")
```

```
eval_new <- eval_new %>%
  mutate(TARGET = as.integer(Predicted))
```

```
train_new1$Predicted <- predictions <- predict(pois2, newdata = train_new1, type = "response")
```

```
train_new1 <- train_new1 %>%
  mutate(TARGET = as.integer(Predicted))
```

```
qplot(train_new1$TARGET, geom = "bar")
```

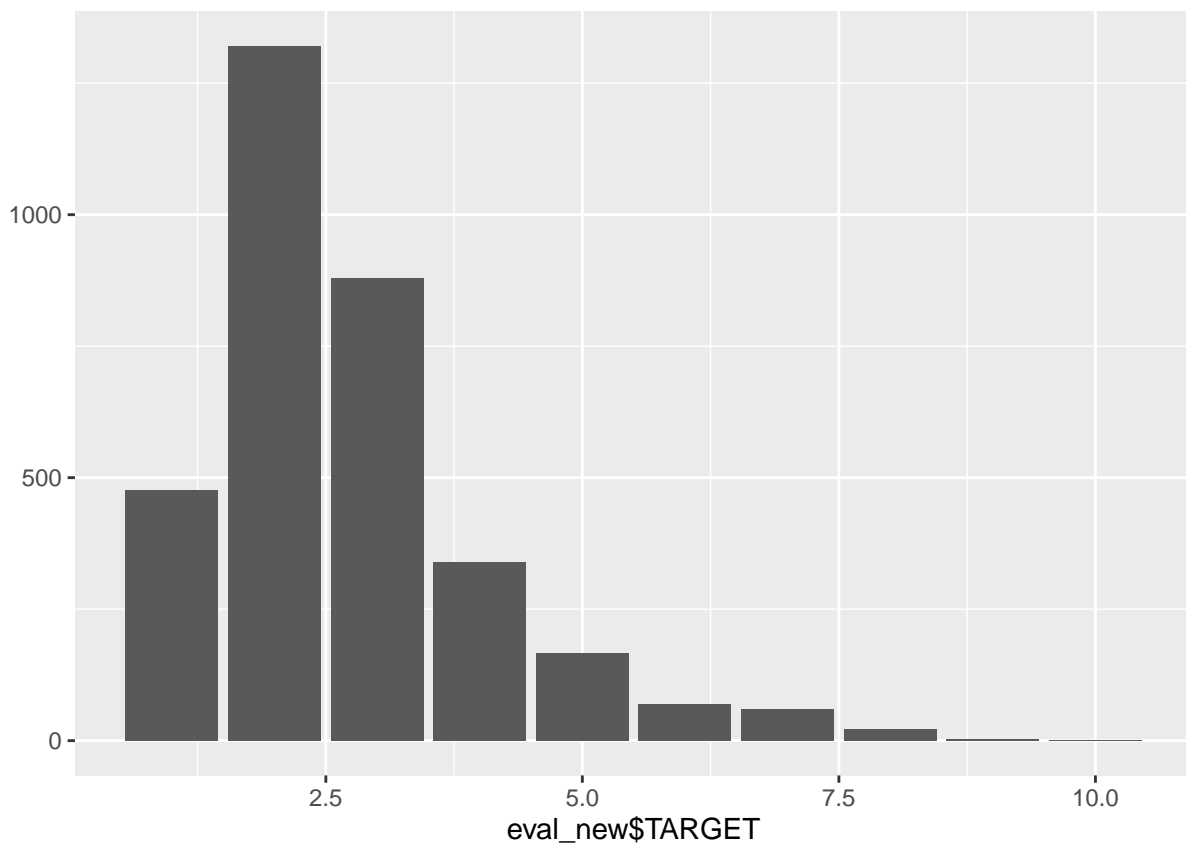


```
# prd1 <- predict(pois2, se.fit=TRUE)
# prd1$se.fit
#
# prd2 <- predict(mlr2, se.fit=TRUE)
# prd2$se.fit

chi_stat <- -2*(logLik(pois1)-logLik(pois2))
1-pchisq(chi_stat, 1)
```

```
## 'log Lik.' 0 (df=6)
```

```
qplot(eval_new$TARGET, geom = "bar")
```



```
summary(eval_new)
```

```
##          IN          TARGET      FixedAcidity      VolatileAcidity
##  Min.   :    3   Min.   : 1.000   Min.   : -18.200   Min.   : -2.8300
##  1st Qu.: 4018   1st Qu.: 2.000   1st Qu.:  5.200   1st Qu.:  0.0800
##  Median : 7906   Median : 2.000   Median :  6.900   Median :  0.2800
##  Mean   : 8048   Mean   : 2.691   Mean   :  6.864   Mean   :  0.3103
##  3rd Qu.:12061   3rd Qu.: 3.000   3rd Qu.:  9.000   3rd Qu.:  0.6300
##  Max.   :16130   Max.   :10.000   Max.   : 33.500   Max.   :  3.6100
##  CitricAcid  ResidualSugar      Chlorides      FreeSulfurDioxide
##  Min.   : -3.1200   Min.   : -128.300   Min.   : -1.15000   Min.   : -563.00
```

```
## 1st Qu.: 0.0000 1st Qu.: 0.500 1st Qu.: 0.02400 1st Qu.: 5.00
## Median : 0.3100 Median : 4.600 Median : 0.04800 Median : 32.00
## Mean : 0.3124 Mean : 5.319 Mean : 0.06143 Mean : 34.95
## 3rd Qu.: 0.6050 3rd Qu.: 15.525 3rd Qu.: 0.14350 3rd Qu.: 70.00
## Max. : 3.7600 Max. : 145.400 Max. : 1.26300 Max. : 617.00
## TotalSulfurDioxide Density pH Sulphates
## Min. : -769.0 Min. : 0.8898 Min. : 0.600 Min. : -3.0700
## 1st Qu.: 32.0 1st Qu.: 0.9883 1st Qu.: 2.990 1st Qu.: 0.3600
## Median : 123.4 Median : 0.9946 Median : 3.220 Median : 0.5300
## Mean : 123.4 Mean : 0.9947 Mean : 3.237 Mean : 0.5346
## 3rd Qu.: 201.0 3rd Qu.: 1.0005 3rd Qu.: 3.460 3rd Qu.: 0.7550
## Max. : 1004.0 Max. : 1.0998 Max. : 6.210 Max. : 4.1800
## Alcohol LabelAppeal AcidIndex STARS
## Min. : -4.20 Min. : -2.00000 Min. : 5.000 Min. : 1.00
## 1st Qu.: 9.10 1st Qu.: -1.00000 1st Qu.: 7.000 1st Qu.: 2.00
## Median : 10.58 Median : 0.00000 Median : 8.000 Median : 2.00
## Mean : 10.58 Mean : 0.01349 Mean : 7.748 Mean : 2.04
## 3rd Qu.: 12.40 3rd Qu.: 1.00000 3rd Qu.: 8.000 3rd Qu.: 2.04
## Max. : 25.60 Max. : 2.00000 Max. : 17.000 Max. : 4.00
## CombinedAcidity CombinedSulfurDioxide Predicted
## Min. : -18.605 Min. : -916.0 Min. : 1.099
## 1st Qu.: 5.175 1st Qu.: 28.0 1st Qu.: 2.280
## Median : 7.230 Median : 159.9 Median : 2.898
## Mean : 7.174 Mean : 159.9 Mean : 3.196
## 3rd Qu.: 9.485 3rd Qu.: 302.0 3rd Qu.: 3.670
## Max. : 33.160 Max. : 1424.0 Max. : 10.298
```

```
# Get AIC values
```

```
aic_mlr1 <- AIC(mlr1)
aic_mlr2 <- AIC(mlr2)
aic_nb1 <- AIC(nb1)
aic_nb2 <- AIC(nb2)
aic_pois1 <- AIC(pois1)
aic_pois2 <- AIC(pois2)
```

```
# Combine into a data frame
```

```
aic_values <- data.frame(
  Model = c("mlr1", "mlr2", "nb1", "nb2", "pois1", "pois2"),
  AIC = c(aic_mlr1, aic_mlr2, aic_nb1, aic_nb2, aic_pois1, aic_pois2)
)
```

```
# View the data frame
```

```
aic_values
```

```
## Model AIC
## 1 mlr1 45401.50
## 2 mlr2 45277.08
## 3 nb1 48235.95
## 4 nb2 48223.34
## 5 pois1 48233.73
## 6 pois2 47749.02
```

```

# Install if necessary
# install.packages("xtable")

# Load the xtable library
library(xtable)

# Convert the data frame to a LaTeX table
latex_table <- xtable(aic_values)

# Print the LaTeX code
print(latex_table, include.rownames = FALSE)

## % latex table generated in R 4.3.2 by xtable 1.8-4 package
## % Thu Mar 7 09:10:42 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{lr}
## \hline
## Model & AIC \\
## \hline
## mlr1 & 45401.50 \\
## mlr2 & 45277.08 \\
## nb1 & 48235.95 \\
## nb2 & 48223.34 \\
## pois1 & 48233.73 \\
## pois2 & 47749.02 \\
## \hline
## \end{tabular}
## \end{table}

```