ST495 FAERS Group Project Final Report

Ryan Mersereau, Carina Fu, Daiwik Bommireddipally, Sydney Chau

9 December 2023

Executive Summary

Our group analyzed the FAERS dataset, which stands for the FDA (Food & Drug Administration) Adverse Event Reporting System. The primary goal of this project was to create several classification models for this dataset using the given response variables dechal and rechal, short for dechallenge and rechallenge. We hoped to use methods and models learned over the semester to accurately predict these response variables. Additionally, this project gave us more experience with data cleaning, manipulation, modeling, and reporting. We found 12 predictor variables to be significant predictors of our response variables, as described in our logistic regression analysis. These variables were age, age_cod, sex, wt, wt_cod, to_mfr, occp_cod, reporter_country, occr_country, role_cod, route, and outc_cod. The pruned classification tree for Dechal using 2020 Q4 data had 4 terminal nodes, a residual mean deviance of 1.702, and a misclassification error rate of 0.418. The pruned tree for Rechal using 2020 Q4 data had 3 terminal nodes, a residual mean deviance of 0.9726, and a misclassification error rate of 0.156. The occp_cod variable (reporter occupation) equal to “consumer” is the most important predictor of the outcome of rechal and dechal, as it is the first split in both trees. The KNN models also proved to be the best predictors of rechal and dechal compared with LDA, QDA, and Naive Bayes. With k=5, using 2020 Q4 for rechalY had a predictive accuracy of 90.60%, and rechalN had a predictive accuracy of 95.23%. The k-nearest neighbors algorithm analysis model for dechalY using 2020 Q4 for dechalY had a predictive accuracy of 61.72%, and dechalN had a predictive accuracy of 76.95%.

Problem Setting

The FAERS dataset contains information on medication errors and adverse event reports. “In response to the COVID-19 pandemic, the FDA launched the FAERS Public Dashboard for COVID-19 emergency use authorization (EUA) products. The COVID-19 EUA FAERS Public

Dashboard provides weekly updates of adverse event reports submitted to FAERS for drugs and therapeutic biological products used under EUA in COVID-19", as seen from the FDA website. Observations in this dataset are self-reported by healthcare professionals, consumers, and manufacturers, as indicated by the reporter occupation variable. It was designed to support the FDA's post-marketing surveillance system and is used to evaluate safety concerns and create regulatory actions to improve product safety and public health.

Although the FAERS dataset was launched in response to the COVID-19 pandemic, Adverse event tracking and data collection have gone on long before the pandemic in 2020. Therefore, we wanted to contrast data from before and after the start of the COVID-19 pandemic. We focused our analysis on data from Q4 of 2018 and 2020, taking a subset of the data to be analyzed, and splitting it into training and testing data for model building. Our response variables of interest from this data are dechal and rechal, and we needed to determine what variables were useful in predicting dechal and rechal. In doing so, we would be able to construct thorough classification trees and other predictive models. This data analysis will give us insight into what factors are associated with patient reactions from stopping or continuing drug therapy, and what differences there are in the data and predictive accuracy of the response variables before and after COVID-19.

Data Description

We used 2020 Q4 and 2018 Q4 FAERS data for our analysis. To obtain the data for these quarters, we downloaded two zip files, each containing seven main .txt files in ASCII format, as shown below:

- DEMOyyQq.txt contains patient demographic and administrative information, a single record for each event report.
- DRUGyyQq.txt contains drug/biologic information for as many medications as were reported for the event (1 or more per event).
- REACyyQq.txt contains all "Medical Dictionary for Regulatory Activities" (MedDRA) terms coded for the adverse event (1 or more)
- OUTCyyQq.txt contains patient outcomes for the event (0 or more).
- RPSRyyQq.txt contains report sources for the event (0 or more).

- THERyyQq.txt contains drug therapy start dates and end dates for the reported drugs (0 or more per drug per event).
- INDIyyQq.txt contains all "Medical Dictionary for Regulatory Activities" (MedDRA) terms coded for the indications for use (diagnoses) for the reported drugs (0 or more per drug per event).

ASCII format files are delimited using '$', and have unique identifiers for the data source, as seen with the last four letters of the data names. Our response variables of interest are dechal and rechal. Dechal stands for the dechallenge code, indicating if the reaction was abated when drug therapy was stopped. Dechal is a categorical variable and can take the values of Y - meaning a positive dechallenge, N - meaning a negative dechallenge, U - meaning unknown, or D - meaning does not apply. Similarly, Rechal stands for the rechallenge code, indicating if a reaction recurred when drug therapy was restarted. Rechal is also a categorical variable and can take the values of Y - meaning a positive rechallenge, N - meaning a negative rechallenge, U - meaning unknown, or D - meaning does not apply. The dataset contains 51 variables in total, with more information on each variable and examples found within the Data Dictionary. We were tasked with deciding which variables from the ASCII files would be useful in predicting our response variables.

Methods

Data extracting and cleaning for the 2018 and 2020 data were done nearly identically. First, the 2020 Q4 data were downloaded from the NBER website. Our first challenge was merging the different data files and cleaning the data to be properly used for modeling. This was done in R using the *dplyr* and *readr* libraries. We read in the seven .txt files, trimming the white spaces of the data, specifying the delimiter, and removing the "caseid" variable because it is repetitive. We then filtered to find the unique primaryid for each of these seven datasets and merged the datasets into a data frame. These datasets were joined using the common primaryid of each observation across all seven datasets. Our final step of data cleaning was removing rows with duplicate primary IDs, keeping only the first appearance. We then wrote this merged data to a .csv file to be used for future analysis.

Before beginning any model building, we needed to conduct preliminary screening and narrow our pool of candidate predictor variables. This was done by immediately excluding variables with only missing (NA) values using the *is.na()* function and dropping these columns from the merged data table. We then added a flag to identify rows where our response variables dechal and rechal were Y or N, as these are the most important observations. We kept 12 predictor variables to be used for modeling. These variables were as follows:

- age: Patient age
- age_cod: Unit abbreviation for patient age
- sex: Patient sex
- wt: Numeric value of patient weight
- wt_cod: Unit abbreviation for patient weight
- to_mfr: Whether (Y/N) voluntary reporter also notified the manufacturer
- occp_cod: Abbreviation for the reporter's type of occupation
- reporter_country: The country of the reporter
- occr_country: The country where the event occurred
- role_cod: Code for drugs reported role in the event
- route: The route of drug administration
- outc_cod: Code for patient outcome

Next, we planned to generate several predictive models using some or all of these variables. We began by creating two logistic regression models using these predictor variables for dechal and rechal using the *glm* function, and output the summary of these models for analysis.

First, we performed linear discriminant analysis (LDA) on the data, creating four predictive tables for predicting rechal in 2018, rechal in 2020, dechal in 2018, and dechal in 2020. This was done by splitting the data into training and testing data, using a split of 80% and 20%, respectively. We also removed one row of data where rechal was equal to ‘U’ to simplify the model and increase predictive accuracy. We were only able to use three variables to create these models, being sex, to_mfr, and occp_cod. This analysis was performed using the *MASS* library, and the *lda* function in R to create models and predictions on the testing data. Then, a confusion matrix was outputted to observe the accuracy of the predictions.

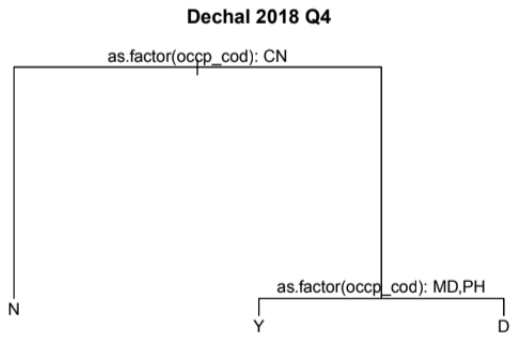
We then performed quadratic discriminant analysis (QDA) on the data to see if the data could be better predicted using a quadratic curve as the discriminant between responses. The

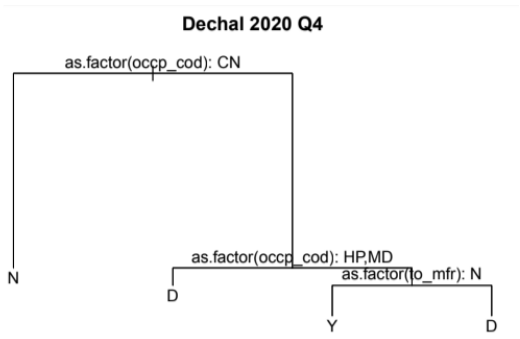
process for generating the QDA models was very similar to as described with LDA, using the same training and test data with the same variables. The accuracy was obtained using the *predict* function and found by taking the mean of predictions compared to the testing data.

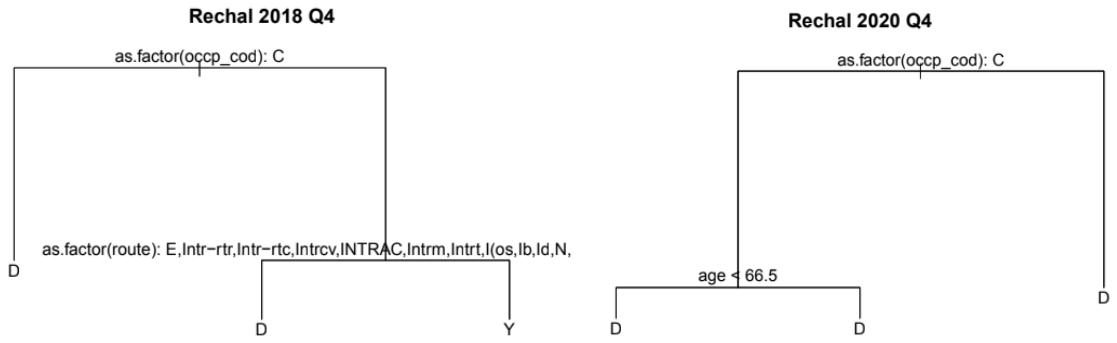
To create the Naive Bayes models, we used the *e1071* library, and a testing and training split of 70% and 30% respectively for the data. As with LDA and QDA, we used the variables sex, to_mfr, and occp_cod, and created a confusion matrix to measure predictive accuracy.

To perform KNN, or K-nearest neighbors, we created a custom R function called knn_find that took parameters for the training data, test data, X, Y, and k values. This would then create a knn model based on the input parameters and generate a prediction with a hopefully high degree of accuracy. We chose to measure the predictive accuracy of the models differently than previously, focusing on whether the response variable (dechal/ rechal) was yes or no. This eliminated the response of 'D', or does not apply, which is not as helpful in interpreting results as Y or N. Therefore, we had 8 predictions instead of 4, with dechal and rechal each being predicted specifically for Y and N, and a binary response.

Finally, we began constructing the classification trees for our response variables. We loaded the libraries *tree* and *cvsm* for tree construction and cross-validation. The trees were constructed and pruned using the optimal tree size obtained from cross-validation, to minimize the misclassification error rate, resulting in four total trees as seen below.







Results

The logistic regression tree for dechal had a null deviance of 5681.2, a residual deviance of 2721.1, an AIC value of 2795.1, and 17 fisher scoring iterations. From the summary output, the variables age, Sex = M, wt, occp_cod = (HP, MD, PH), reporter_country = (GB, US), and role_cod = SS were found to be significant in predicting dechal. The logistic regression tree for rechal had a null deviance of 5817.5, a residual deviance of 2757.5, an AIC value of 2831.7, and 17 fisher scoring iterations. From the summary output, all the variables found to be significant in predicting dechal were significant in predicting rechal, except for wt and role_cod = SS.

The linear discriminant analysis model for rechal using 2018 Q4 training data had a predictive accuracy of 80.94% for predicting rechal to be ‘Y’, ‘N’, or ‘D’. Using 2020 Q4 for rechal had a predictive accuracy of 85.02%. The LDA model for dechal using 2018 Q4 training data had a predictive accuracy of 47.36%. Using 2020 Q4 for dechal had a predictive accuracy of 58.37%.

The quadratic discriminant analysis model for rechal using 2018 Q4 data had a predictive accuracy of 55.28%. Using 2020 Q4 for rechal had a predictive accuracy of 85.02%. The QDA model for dechal using 2018 Q4 training data had a predictive accuracy of 47.36%. Using 2020 Q4 for dechal had a predictive accuracy of 58.37%, Identical to the LDA models' predictive accuracy.

The Naïve Bayes analysis model for rechal using 2018 Q4 data had a predictive accuracy of 80.04%. Using 2020 Q4 for rechal had a predictive accuracy of 85.83%. The Naïve Bayes analysis model for dechal using 2018 Q4 training data had a predictive accuracy of 52.83%.

Using 2020 Q4 for dechal had a predictive accuracy of 54.83%. The predictive accuracy for rechal is much higher than dechal.

The k-nearest neighbors algorithm analysis model with k=5 for rechalY using 2018 Q4 data had a predictive accuracy of 85.10%, and for rechalN using 2018 Q4 data had a predictive accuracy of 94.58%. Using 2020 Q4 for rechalY had a predictive accuracy of 90.60%, and rechalN had a predictive accuracy of 95.23%. The k-nearest neighbors algorithm analysis model for dechalY using 2018 Q4 training data had a predictive accuracy of 57.02%, and dechalN had a predictive accuracy of 65.89%. Using 2020 Q4 for dechalY had a predictive accuracy of 61.72%, and dechalN had a predictive accuracy of 76.95%. The predictive accuracy for rechal is much higher than dechal. The predictive accuracy for N is higher than Y for both variables.

After pruning our classification trees and performing cross-validation to find the optimal tree size we were left with four trees. The pruned tree for Dechal using 2018 Q4 data had 3 terminal nodes, a residual mean deviance of 1.711, and a misclassification error rate of 0.4884. The variables occp_cop = C, MD, and PH were used in constructing the tree. The pruned tree for Dechal using 2020 Q4 data had 4 terminal nodes, a residual mean deviance of 1.702, and a misclassification error rate of 0.418.

The pruned tree for Rechal using 2018 Q4 data had 3 terminal nodes, a residual mean deviance of 1.151, and a misclassification error rate of 0.191. The variables occp_cod = C and route were used in constructing the tree. The pruned tree for Rechal using 2020 Q4 data had 3 terminal nodes, a residual mean deviance of 0.9726, and a misclassification error rate of 0.156. The variables occp_cod = C and age were used in constructing the tree.

Upon first glance at the trees, the residual mean deviance and misclassification error rates for the rechal trees are much lower than the dechal trees. Comparing the dechal trees for 2018 and 2020, the residual mean deviance and misclassification error rates were very similar. For rechal, however, the tree using 2020 data showed significant improvement over the 2018 tree.

Discussion

The models we generated had varying degrees of success in accurately predicting our response variables of dechal and rechal. Linear discriminant analysis was fairly accurate for predicting rechal, but not for predicting dechal, as seen in our results. The quadratic discriminant analysis

produced worse predictive accuracy for predicting 2018 rechal but was identical for all other groups. This leads us to believe that LDA is better for predicting rechal in 2018, but shows no improvement over QDA for other groups. The Naive Bayes analysis gave us very similar results as well, with predictive accuracy for rechal staying around 80%, and predictive accuracy for dechal staying around 50%. Our models did not give us any significant indications that predicting response for 2018 was more or less accurate than for 2020, so we are unable to conclude that COVID-19 affected the data and its predictive accuracy on the response variables.

However, our models for KNN appear to be much better in predicting rechal and dechal than the previous models. We believe that this is due to specifically predicting whether rechal/dechal is 'Yes' or 'No', which removes the predictions of 'D', or does not apply. The model predictions for when our response variables are D are likely where much of the inaccuracy comes from, therefore the model is much more accurate when it only has to distinguish between yes and no.

For both the dechal and rechal classification trees, the variable at the top of the tree is occp_cod: CN, which correlates to the reporter's type of occupation, where CN is a consumer. This means that the occp_cod variable equal to "consumer" is the most important predictor of the outcome of rechal and dechal. Occp_cod is the only variable used in the construction of the 2018 dechal tree. For the 2020 dechal tree, occp_cod and to_mfr are used in the tree construction. Occp_cod and route are used in the 2018 rechal tree, and occp_cod and age are used in the 2020 rechal tree.

From the results, there is a lower mean deviance and misclassification error rate for rechal compared to dechal, suggesting the classification trees for rechal are more accurate. However, some of the terminal nodes in both trees end in 'D', or 'does not apply'. Specifically, 1 in 3 of the 2018 dechal nodes end in D,2 of 4 of the 2020 dechal nodes, 2 of 3 of the 2018 rechal nodes, and 3 of 3 of the 2020 rechal nodes. This is less conclusive than a yes or no answer, so it may be harder to conclude the specific effectiveness of variables on rechallenge code from this classification tree.

While working on this project, we encountered several challenges that impeded our ability to effectively analyze the data. The first major challenge we encountered was working with a very large amount of data. Our dataset had over 50 variables and over 100 million observations. Due to this large size, it was challenging to select good predictor variables, as well

as clean and merge the data. Many of the variables were difficult to work with. For example, rept_cod and mf_sndr only have one unique observed value, whereas start_dt has more than 20. We also encountered some problems with missing values and had to figure out how to work with observations that contained partially complete data. We initially wanted to use data from 2023 but ran into challenges with incomplete data, so we had to go further back to 2018 and 2020.

Some issues we ran into while building our models included our logistic regression models running but not converging, which may have slightly affected the accuracy of the regression line. Additionally, many of the variables needed to be removed from our models, leaving us with only three variables.

We believe further improvements could be made if we included more predictor variables in our models, especially the classification tree. This would allow us to more accurately predict dechal and rechal, and see which variables are most influential in predicting them. Limiting the prediction of rechal and dechal to only Y and N could also improve the accuracy and interpretative power of our models. We did this with KNN and saw that the predictive accuracy dramatically increased, in some cases to over 90%. Expanding this idea to our other models could yield improved accuracy.

References

FAERS Website:
https://www.fda.gov/drugs/questions-and-answers-fdas-adverse-event-reporting-system-faers/fda-adverse-event-reporting-system-faers-public-dashboard

FAERS Data Dictionary:
https://pharmahub.org/app/site/resources/2018/01/00739/FDA-FAERS-Data-Dictionary.pdf

NBER Data Download:
https://www.nber.org/research/data/fda-adverse-event-reporting-system