Predicting Genetic Disorders

Emma Oo[1], Sindhu Bhattarai[2], Dave Friesen[3]

06/27/2022

Objective and Hypothesis

[. . .]

Data Load and Validation

```
# Load dataset(s)
gd_df <- read.csv("../data/train_genetic_disorders.csv", header = TRUE)

# Data validation and understanding, including structure, content, and statistical characteristics covered below
```

Data Structure Review

```
# Summarize base dataset and [optionally] sample rows
str(gd_df)
```

```
'data.frame':   22083 obs. of  45 variables:
 $ Patient.Id                                    : chr  "PID0x6418" "PID0x25d5" "PID0x4a82" "PID0x4ac8" ...
 $ Patient.Age                                   : int  2 4 6 12 11 14 3 3 11 4 ...
 $ Genes.in.mother.s.side                        : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Inherited.from.father                         : chr  "No" "Yes" "No" "No" ...
 $ Maternal.gene                                 : chr  "Yes" "No" "No" "Yes" ...
 $ Paternal.gene                                 : chr  "No" "No" "No" "No" ...
 $ Blood.cell.count..mcL.                        : num  4.76 4.91 4.89 4.71 4.72 ...
 $ Patient.First.Name                            : chr  "Richard" "Mike" "Kimberly" "Jeffery" ...
 $ Family.Name                                   : chr  "" "" "" "Hoelscher" ...
 $ Father.s.name                                 : chr  "Larre" "Brycen" "Nashon" "Aayaan" ...
 $ Mother.s.age                                  : int  NA NA 41 21 32 NA NA 40 45 44 ...
 $ Father.s.age                                  : int  NA 23 22 NA NA NA 63 NA 44 42 ...
 $ Institute.Name                                : chr  "Boston Specialty & Rehabilitation Hospital" "St. Margaret's Ho
spital For Women" "" "" ...
 $ Location.of.Institute                         : chr  "55 FRUIT ST\nCENTRAL, MA 02114\n(42.36247485742686, -71.069247
24545246)" "1515 COMMONWEALTH AV\nALLSTON/BRIGHTON, MA 02135\n(42.34665771451756, -71.14136122385321)" "-" "55 FRUIT ST\n
CENTRAL, MA 02114\n(42.36247485742686, -71.06924724545246)" ...
 $ Status                                        : chr  "Alive" "Deceased" "Alive" "Deceased" ...
 $ Respiratory.Rate..breaths.min.                : chr  "Normal (30-60)" "Tachypnea" "Normal (30-60)" "Tachypnea" ...
 $ Heart.Rate..rates.min                         : chr  "Normal" "Normal" "Tachycardia" "Normal" ...
 $ Test.1                                        : int  0 NA 0 0 0 0 NA 0 0 0 ...
 $ Test.2                                        : int  NA 0 0 0 0 0 0 0 0 0 ...
 $ Test.3                                        : int  NA 0 0 0 0 0 0 NA 0 0 ...
 $ Test.4                                        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Test.5                                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Parental.consent                              : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Follow.up                                     : chr  "High" "High" "Low" "High" ...
 $ Gender                                        : chr  "" "" "" "Male" ...
 $ Birth.asphyxia                                : chr  "" "No" "No record" "Not available" ...
 $ Autopsy.shows.birth.defect..if.applicable.    : chr  "Not applicable" "None" "Not applicable" "No" ...
 $ Place.of.birth                                : chr  "Institute" "" "" "Institute" ...
 $ Folic.acid.details..peri.conceptional.        : chr  "No" "Yes" "Yes" "No" ...
 $ H.O.serious.maternal.illness                  : chr  "" "Yes" "No" "Yes" ...
 $ H.O.radiation.exposure..x.ray.                : chr  "No" "Not applicable" "Yes" "-" ...
 $ H.O.substance.abuse                           : chr  "No" "Not applicable" "" "Not applicable" ...
 $ Assisted.conception.IVF.ART                   : chr  "No" "No" "Yes" "" ...
 $ History.of.anomalies.in.previous.pregnancies  : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ No..of.previous.abortion                      : int  NA NA 4 1 4 0 3 1 0 1 ...
 $ Birth.defects                                 : chr  "" "Multiple" "Singular" "Singular" ...
 $ White.Blood.cell.count..thousand.per.microliter.: num  9.86 5.52 NA 7.92 4.1 ...
 $ Blood.test.result                             : chr  "" "normal" "normal" "inconclusive" ...
 $ Symptom.1                                     : int  1 1 0 0 0 1 0 0 1 0 ...
 $ Symptom.2                                     : int  1 NA 1 0 0 0 0 0 1 0 ...
 $ Symptom.3                                     : int  1 1 1 1 0 0 0 1 1 1 ...
 $ Symptom.4                                     : int  1 1 1 0 0 1 0 NA 0 1 ...
 $ Symptom.5                                     : int  1 0 1 0 NA 0 0 0 1 1 ...
 $ Genetic.Disorder                              : chr  "Mitochondrial genetic inheritance disorders" "" "Multifactoria
l genetic inheritance disorders" "Mitochondrial genetic inheritance disorders" ...
 $ Disorder.Subclass                             : chr  "Leber's hereditary optic neuropathy" "Cystic fibrosis" "Diabet
```

```
es" "Leigh syndrome" ...
```

```
#head(gd_df, 3)
```

Preliminary Feature Reduction (clearly n/a to Objective and Hypothesis)

```
# Define n/a columns and subset dataframe; Note retaining "some" informational variables like "Institute.Name" for
#   possible descriptive analytic purposes
drop_cols <- c("Patient.First.Name",
               "Family.Name",
               "Father.s.name",
               "Institute.Name",
               "Location.of.Institute")
gd_df <- gd_df[ , !(names(gd_df) %in% drop_cols)]
```

Class Target and Label Review

```
# Check for missing labels; set aside where missing
missing_target <- which(is.na(gd_df$Disorder.Subclass) | (gd_df$Disorder.Subclass == ""))
cat("Rows pre-subset for missing labels: ", format(nrow(gd_df), format = "d", big.mark = ","), sep = "")
```

```
Rows pre-subset for missing labels: 22,083
```

```
gd_hold_df <- gd_df[missing_target, ]
gd_df <- gd_df[-missing_target, ]
cat("Held rows with missing labels: ", format(nrow(gd_hold_df), format = "d", big.mark = ","), sep = "")
```

```
Held rows with missing labels: 3,140
```

```
cat("Net rows (labeled): ", format(nrow(gd_df), format = "d", big.mark = ","), sep = "")
```

```
Net rows (labeled): 18,943
```

```
# Show frequency distribution for [prospective] target class(es)
show_frequency <- function(desc, c) {
  t <- as.data.frame(prop.table(table(c)))
  colnames(t) <- c("Class", "Frequency")
  cat(desc, "\n"); print(t[order(-t$Freq, t$Class), 1:2], row.names = FALSE)
}
show_frequency("Pre-Split Frequency Distribution", gd_df$Disorder.Subclass)
```

```
Pre-Split Frequency Distribution
                             Class Frequency
                      Leigh syndrome     0.258
              Mitochondrial myopathy     0.222
                     Cystic fibrosis     0.173
                           Tay-Sachs     0.142
                            Diabetes     0.092
                      Hemochromatosis     0.068
 Leber's hereditary optic neuropathy     0.032
                          Alzheimer's     0.008
                               Cancer     0.005
```

```
# Move the target class to "top" of dataframe so column removals don't impact
gd_df <- gd_df[ , c(ncol(gd_df), 1:(ncol(gd_df) - 1))]
target_col = 1



gd_df$Disorder.Subclass <- gsub("'", "", gd_df$Disorder.Subclass, fixed = TRUE)
gd_df$Disorder.Subclass <- gsub(" ", ".", gd_df$Disorder.Subclass, fixed = TRUE)
gd_df$Disorder.Subclass <- gsub("-", ".", gd_df$Disorder.Subclass, fixed = TRUE)
```

Data Splitting

```
# Split data 80/20 train/test, using caret's inherent stratified split to compensate for class imbalance
set.seed(1)
```

```
train_index <- createDataPartition(gd_df$Disorder.Subclass, times = 1, p = 0.80, list = FALSE)
train_df <- gd_df[train_index, ]
test_df <- gd_df[-train_index, ]
show_frequency("Post-Split Frequency Distribution (Train)", train_df$Disorder.Subclass)
```

```
Post-Split Frequency Distribution (Train)
                              Class Frequency
                     Leigh.syndrome     0.258
                Mitochondrial.myopathy     0.222
                       Cystic.fibrosis     0.173
                             Tay.Sachs     0.142
                              Diabetes     0.092
                       Hemochromatosis     0.068
 Lebers.hereditary.optic.neuropathy     0.032
                            Alzheimers     0.008
                                Cancer     0.005
```

Data Cleaning (and reduction)

Data (Sample) Characteristic Review for Pre-Processing

(Suppressing custom code for simplicity)

```
# Generate a summary (cursory) view of base dataset for initial understanding and pre-processing direction
univariate(train_df)
```

```
Summary Univariate Analysis (15,158 observations)
                    Type          NA BlankZ  Unique     Min     Max    Mean  Median Outlier< >Outlier Kurtosis Skewness
Disorder.Subclass   character                  9
Patient.Id          character                 15,158
Patient.Age         integer       6%     6%      15              14              7    No       Yes       0.017   -1.211
Genes.in.mother.s... character                  2
Inherited.from.fa... character              1%      3
Maternal.gene       character              12%      3
Paternal.gene       character                  2
Blood.cell.count.... numeric                15,158   4.093   5.610   4.900   4.902    No       Yes      -0.011   -0.037
Mother.s.age        integer      26%           34      18      51              35    No       Yes      -0.006   -1.219
Father.s.age        integer      25%           45      20      64              42    No       Yes      -0.002   -1.210
Status              character                  2
Respiratory.Rate.... character              9%      3
Heart.Rate..rates... character              9%      3
Test.1              integer       9%    90%      1                                  No       No
Test.2              integer       9%    90%      1                                  No       No
Test.3              integer       9%    90%      1                                  No       No
Test.4              integer       9%             1       1       1               1    No       No
Test.5              integer       9%    90%      1                                  No       No
Parental.consent    character              9%      2
Follow.up           character              9%      3
Gender              character              9%      4
Birth.asphyxia      character              9%      5
Autopsy.shows.bir... character              4%      5
Place.of.birth      character              9%      3
Folic.acid.detail... character              9%      3
H.O.serious.mater... character              8%      3
H.O.radiation.exp... character              9%      5
H.O.substance.abuse character               9%      5
Assisted.concepti... character              9%      3
History.of.anomal... character              9%      3
No..of.previous.a... integer       9%    18%      5               4               2    No       Yes       0.001   -1.292
Birth.defects       character              9%      3
White.Blood.cell.... numeric       9%          11,858   3.000  12.000   7.460   7.443    No       Yes       0.020   -0.979
Blood.test.result   character              9%      5
Symptom.1           integer       9%    37%      2               1               1    No       Yes      -0.369   -1.864
Symptom.2           integer       9%    40%      2               1               1    No       Yes      -0.197   -1.961
Symptom.3           integer       8%    41%      2               1               1    No       Yes      -0.166   -1.973
Symptom.4           integer       9%    45%      2               1                    No       Yes       0.010   -2.000
Symptom.5           integer       9%    48%      2               1                    No       Yes       0.146   -1.979
Genetic.Disorder    character              9%      4
```

Missing Values

```
# Genes.in.mother.s.side, Paternal.gene, Blood.cell.count..mcL., Status - n/a

# Impute basic integer values with medians
medianf <- function(x) {
```

```
 result <- median(x, na.rm = TRUE)
 if (is.integer(x))
   result <- as.integer(result)
 return(result)
}
median_cols = c("Patient.Age", "Mother.s.age", "Father.s.age", "No..of.previous.abortion")
for (n in median_cols) {
  train_df[n][is.na(train_df[n])] <- apply(train_df[n], 2, medianf)
  test_df[n][is.na(test_df[n])] <- apply(test_df[n], 2, medianf)
}

# Impute categorical blanks with common "notprovided"; note we could also impute these with categorical mode,
#   or most frequent categorical value of each column using the cmode() function below
cols_tofill <- c("Inherited.from.father",
                 "Maternal.gene",
                 "Respiratory.Rate..breaths.min.",
                 "Heart.Rate..rates.min",
                 "Parental.consent",
                 "Follow.up",
                 "Gender",
                 "Birth.asphyxia",
                 "Autopsy.shows.birth.defect..if.applicable.",
                 "Place.of.birth",
                 "Folic.acid.details..peri.conceptional.",
                 "H.O.serious.maternal.illness",
                 "H.O.radiation.exposure..x.ray.",
                 "H.O.substance.abuse",
                 "Assisted.conception.IVF.ART",
                 "History.of.anomalies.in.previous.pregnancies",
                 "Birth.defects",
                 "Blood.test.result")
train_df[cols_tofill][train_df[cols_tofill] == ""] <- "notprovided"
test_df[cols_tofill][test_df[cols_tofill] == ""] <- "notprovided"

cmode <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

# Impute what appear to be masked "flag" columns iwth placeholder -1 values. . .
flag_cols <- c("Test.1", "Test.2", "Test.3", "Test.4", "Test.5",
               "Symptom.1", "Symptom.2", "Symptom.3", "Symptom.4", "Symptom.5")
train_df[flag_cols][is.na(train_df[flag_cols])] <- as.integer(-1)
test_df[flag_cols][is.na(test_df[flag_cols])] <- as.integer(-1)

# Impute mean for one numeric column
train_df$White.Blood.cell.count..thousand.per.microliter.[is.na(train_df$White.Blood.cell.count..thousand.per.microliter.
)] <-
  mean(train_df$White.Blood.cell.count..thousand.per.microliter., na.rm = TRUE)
test_df$White.Blood.cell.count..thousand.per.microliter.[is.na(test_df$White.Blood.cell.count..thousand.per.microliter.)]
 <-
  mean(test_df$White.Blood.cell.count..thousand.per.microliter., na.rm = TRUE)

# Note not using knnImpute for the limited number of numerical [prospective] features given that it
#   centers/scales, which is illogical for the values in this dataset
#pp <- preProcess(train_df[ , -target_col, drop = FALSE], method = "knnImpute", k = 10)
#train_df[ , -target_col] <- predict(pp, train_df[ , -target_col, drop = FALSE])
#test_df[ , -target_col] <- predict(pp, test_df[ , -target_col, drop = FALSE])

# Last on the list: Genetic.Disorder - we're not classifying to this but it is relevant/informational as a
#   superclass to the target Disorder.Subclass and shuold ultimately be imputed using similar Disorder.Subclass
#   observations which do have valid Genetic.Disorder values
```

Feature Updates (including variable types/formats, names)

```
# Re-type variables
factor_cols <- c("Genes.in.mother.s.side",
                 "Inherited.from.father",
                 "Maternal.gene",
                 "Paternal.gene",
                 "Status",
                 "Respiratory.Rate..breaths.min.",
                 "Heart.Rate..rates.min",
                 "Parental.consent",
                 "Follow.up",
                 "Gender",
                 "Birth.asphyxia",
                 "Autopsy.shows.birth.defect..if.applicable.",
                 "Folic.acid.details..peri.conceptional.",
                 "H.O.serious.maternal.illness",
                 "H.O.radiation.exposure..x.ray.",
                 "H.O.substance.abuse",
```

```
                  "Assisted.conception.IVF.ART",
                  "History.of.anomalies.in.previous.pregnancies",
                  "Birth.defects",
                  "Blood.test.result",
                  "Disorder.Subclass")
train_df[factor_cols] <- lapply(train_df[factor_cols], factor)
test_df[factor_cols] <- lapply(test_df[factor_cols], factor)
# Note dummy variables may be introduced below for e.g., logistic regression

# Simplify variable naming
# [TBD]

# Generate updated summary of base dataset
univariate(train_df)
```

```
Summary Univariate Analysis (15,158 observations)
                     Type          NA BlankZ  Unique     Min     Max    Mean  Median Outlier< >Outlier Kurtosis Skewness
Disorder.Subclass    factor                       9
Patient.Id           character                15,158
Patient.Age          integer               6%      15              14               7    No       Yes       0.016   -1.090
Genes.in.mother.s... factor                       2
Inherited.from.fa... factor                       3
Maternal.gene        factor                       3
Paternal.gene        factor                       2
Blood.cell.count.... numeric                  15,158   4.093   5.610   4.900   4.902    No       Yes      -0.011   -0.037
Mother.s.age         integer                     34      18      51              35    No       Yes      -0.048   -0.593
Father.s.age         integer                     45      20      64              42    No       Yes      -0.007   -0.600
Status               factor                       2
Respiratory.Rate.... factor                       3
Heart.Rate..rates... factor                       3
Test.1               integer              90%     2      -1                              No       No       -2.786    5.762
Test.2               integer              90%     2      -1                             Yes       No       -2.791    5.792
Test.3               integer              90%     2      -1                             Yes       No       -2.851    6.130
Test.4               integer                      2      -1       1              1    No       No       -2.798    5.829
Test.5               integer              90%     2      -1                              No       No       -2.799    5.837
Parental.consent     factor                       2
Follow.up            factor                       3
Gender               factor                       4
Birth.asphyxia       factor                       5
Autopsy.shows.bir... factor                       5
Place.of.birth       character                    3
Folic.acid.detail... factor                       3
H.O.serious.mater... factor                       3
H.O.radiation.exp... factor                       5
H.O.substance.abuse  factor                       5
Assisted.concepti... factor                       3
History.of.anomal... factor                       3
No..of.previous.a... integer              18%     5               4               2    No       Yes                -1.116
Birth.defects        factor                       3
White.Blood.cell.... numeric                  11,859   3.000  12.000   7.460   7.460    No       Yes       0.021   -0.768
Blood.test.result    factor                       5
Symptom.1            integer              37%     3      -1       1              1    No       Yes      -0.769   -0.496
Symptom.2            integer              40%     3      -1       1                   No       Yes      -0.643   -0.624
Symptom.3            integer              41%     3      -1       1                   No       Yes      -0.626   -0.613
Symptom.4            integer              45%     3      -1       1                   No       Yes      -0.502   -0.679
Symptom.5            integer              48%     3      -1       1                   No       Yes      -0.413   -0.702
Genetic.Disorder     character             9%     4
```

Zero/Near-Zero Variances

```
# n/a for this dataset
```

Duplicate Values

```
# n/a for this dataset
```

“Noisy” Data

```
# n/a for this dataset
```

Data Transformation

Centering/Scaling (standardizing/normalizing)

```
# n/a for this dataset?
```

Statistical Characteristics (including distribution, skewness, outliers)

```
#summary(train_df)
```

Other Feature Engineering (transformation, aggregation, enrichment)

```
# n/a for this dataset?
```

Multivariate Analysis (and reduction)

Collinearity and Dependencies

Predictor Transformations (e.g., PCA)

Modeling

Feature Selection

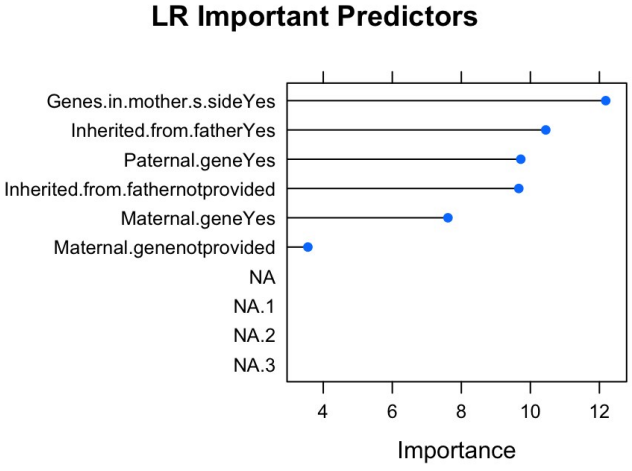
Training, Testing (validating), and Evaluation (iteration n)

```
# Create common control for models and set predictor set
set.seed(1)
fit_control <- trainControl(method = "cv",
                            number = 10,
                            savePredictions = "all",
                            classProbs = TRUE,
                            summaryFunction = multiClassSummary)
predictor_cols <- c("Genes.in.mother.s.side",
                    "Inherited.from.father",
                    "Maternal.gene",
                    "Paternal.gene")
#Patient.Age
#*Genes.in.mother.s.side
#*Inherited.from.father
#*Maternal.gene
#*Paternal.gene
#Blood.cell.count..mcL.
#Mother.s.age
#Father.s.age
#Status
#Respiratory.Rate..breaths.min.
#Heart.Rate..rates.min
#Test.1
#Test.2
#Test.3
#Test.4
#Test.5
#Parental.consent
#Follow.up
#Gender
#Birth.asphyxia
#Autopsy.shows.birth.defect..if.applicable.
#Place.of.birth
#Folic.acid.details..peri.conceptional.
#H.O.serious.maternal.illness
#H.O.radiation.exposure..x.ray.
#H.O.substance.abuse
#Assisted.conception.IVF.ART
#History.of.anomalies.in.previous.pregnancies
#No..of.previous.abortion
#Birth.defects
#White.Blood.cell.count..thousand.per.microliter.
#Blood.test.result
#Symptom.1
```

```r
#Symptom.2
#Symptom.3
#Symptom.4
#Symptom.5

# Train logistic regression model
invisible(capture.output(
  lr_fit <- train(x = train_df[ , predictor_cols, drop = FALSE],
                  y = train_df$Disorder.Subclass,
                  method = "multinom",
                  metric = "ROC",
                  trControl = fit_control)
))
#lr_fit
#lr_fit$finalModel

# Generate confusion matrix
lr_cm <- confusionMatrix(lr_fit, norm = "none")

# Simplify class names for more coherent confusion matrix, and output
for (n in 1:length(rownames(lr_cm$tab)))
  rownames(lr_cm$table)[n] <- paste(rownames(lr_cm$table)[n], " (", AscToChar(64 + n), ")", sep = "")
for (n in 1:length(colnames(lr_cm$tab)))
  colnames(lr_cm$table)[n] <- paste("Class", AscToChar(64 + n))
lr_cm
```

```
Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

                                          Reference
Prediction                                 Class A Class B Class C Class D Class E Class F Class G Class H Class I
  Alzheimers (A)                                 0       0       0       0       0       0       0       0       0
  Cancer (B)                                     0       0       0       0       0       0       0       0       0
  Cystic.fibrosis (C)                           37       0     325     216      17     105     265     135      72
  Diabetes (D)                                   0       0       0       0       0       0       0       0       0
  Hemochromatosis (E)                            0       0       0       0       0       0       0       0       0
  Lebers.hereditary.optic.neuropathy (F)         0       0       0       0       0       0       0       0       0
  Leigh.syndrome (G)                            79      26    1907    1026     563     342    2736    2252    1300
  Mitochondrial.myopathy (H)                     3      48     390     153     453      39     914     975     780
  Tay.Sachs (I)                                  0       0       0       0       0       0       0       0       0

 Accuracy (average) : 0.2663
```

```r
# Plot ROC curve
#lr_roc <- roc(response = lr_fit$pred$obs,
#              predictor = lr_fit$pred$Yes,
#              levels = rev(levels(lr_fit$pred$obs)))
#if (show_output) { plot(lr_roc, legacy.axes = TRUE, main = "LR ROC Curve",
#                        xlab = "Fall Positive Rate (Specificity)",
#                        ylab = "True Positive Rate (Sensitivity)") }
#if (show_output) { lr_roc$auc }

# Check for most important predictors
plot(varImp(lr_fit, scale = FALSE), top = 10, main = "LR Important Predictors")
```




```r
varImp(lr_fit)
```

```
multinom variable importance

                                Overall
Genes.in.mother.s.sideYes         100.0
Inherited.from.fatherYes           79.9
Paternal.geneYes                   71.5
Inherited.from.fathernotprovided   70.8
Maternal.geneYes                   47.0
Maternal.genenotprovided            0.0
```

```
# Establish test_results dataframe
test_results <- data.frame(obs = test_df$Disorder.Subclass,
                           model_2_tbd = 0.0,
                           model_3_tbd = 0.0,
                           model_4_tbd = 0.0,
                           model_5_tbd = 0.0)

# Validate model vs. test data
test_results$lr <- predict(lr_fit, test_df[ , predictor_cols, drop = FALSE])
```

Optimization, Tuning, Selection

---

1. University of San Diego, eoo@sandiego.edu↩
2. University of San Diego, sbhattarai@sandiego.edu↩
3. University of San Diego, dfriesen@sandiego.edu↩