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.Id",
               "Patient.First.Name",
               "Family.Name",
               "Father.s.name",
               "Institute.Name",
               "Location.of.Institute",
               "Status",
               "Test.1",
               "Test.2",
               "Test.3",
               "Test.4",
               "Test.5",
               "Parental.consent",
               "Place.of.birth")
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

# Clean (prelim) target class values
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.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
Respiratory.Rate.... character             9%   3
Heart.Rate..rates... character             9%   3
Follow.up            character             9%   3
Gender               character             9%   4
Birth.asphyxia       character             9%   5
Autopsy.shows.bir... character             4%   5
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",
                 "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")
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("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("Disorder.Subclass",
                 "Genes.in.mother.s.side",
                 "Inherited.from.father",
                 "Maternal.gene",
                 "Paternal.gene",
                 "Respiratory.Rate..breaths.min.",
                 "Heart.Rate..rates.min",
                 "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",
                 "Genetic.Disorder")
```

```
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 (model-dependent)

# Simplify variable naming
rename_cols <- c("Disorder_Subclass",
                 "Patient_Age",
                 "Genes_in_mothers_side",
                 "Inherited_from_father",
                 "Maternal_gene",
                 "Paternal_gene",
                 "Blood_cell_count_mcL",
                 "Mothers_age",
                 "Fathers_age",
                 "Respiratory_Rate_breaths_min",
                 "Heart_Rate_min",
                 "Follow_up",
                 "Gender",
                 "Birth_asphyxia",
                 "Autopsy_shows_birth_defect",
                 "Folic_acid_details_peri_conceptional",
                 "HO_serious_maternal_illness",
                 "HO_radiation_exposure_xray",
                 "HO_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",
                 "Symptom_2",
                 "Symptom_3",
                 "Symptom_4",
                 "Symptom_5",
                 "Genetic_Disorder")
colnames(train_df) <- rename_cols
colnames(test_df) <- rename_cols

# 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_Age          integer             6%      15              14                7    No       Yes       0.016   -1.090
Genes_in_mothers_... factor                   2
Inherited_from_fa... factor                   3
Maternal_gene        factor                   3
Paternal_gene        factor                   2
Blood_cell_count_mcL numeric                15,158   4.093   5.610   4.900   4.902    No       Yes      -0.011   -0.037
Mothers_age          integer                 34      18      51              35    No       Yes      -0.048   -0.593
Fathers_age          integer                 45      20      64              42    No       Yes      -0.007   -0.600
Respiratory_Rate_... factor                   3
Heart_Rate_min       factor                   3
Follow_up            factor                   3
Gender               factor                   4
Birth_asphyxia       factor                   5
Autopsy_shows_bir... factor                   5
Folic_acid_detail... factor                   3
HO_serious_matern... factor                   3
HO_radiation_expo... factor                   5
HO_substance_abuse   factor                   5
Assisted_concepti... factor                   3
History_of_anomal... factor                   3
No_of_previous_ab... 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     factor                   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

```
# Calculate Cramer's V "measure of association" between nominal factor variables (uses Chi-square statistic)
cscorr <- PairApply(train_df[ , sapply(train_df, is.factor)], CramerV, symmetric = TRUE)

# Shorten variable names for ease of reviewing output matrix
rn <- rownames(cscorr)
for (n in 1:length(rownames(cscorr))) {
  rn[n] <- paste(rownames(cscorr)[n], " (", AscToChar(64 + n), ")", sep = "")
  rownames(cscorr)[n] <- paste(AscToChar(64 + n))
}
for (n in 1:length(colnames(cscorr)))
  colnames(cscorr)[n] <- paste(AscToChar(64 + n))

# Show master list of variable names along with output ("correlation") matrix
cat(rn, sep = "\n")
```

```
Disorder_Subclass (A)
Genes_in_mothers_side (B)
Inherited_from_father (C)
Maternal_gene (D)
Paternal_gene (E)
Respiratory_Rate_breaths_min (F)
Heart_Rate_min (G)
Follow_up (H)
Gender (I)
Birth_asphyxia (J)
Autopsy_shows_birth_defect (K)
Folic_acid_details_peri_conceptional (L)
HO_serious_maternal_illness (M)
HO_radiation_exposure_xray (N)
HO_substance_abuse (O)
Assisted_conception_IVF_ART (P)
History_of_anomalies_in_previous_pregnancies (Q)
Birth_defects (R)
Blood_test_result (S)
Genetic_Disorder (T)
```

```
cscorr
```

```
     A     B     C     D     E     F     G    H    I     J    K     L     M     N    O     P     Q     R    S    T
A 1.00 0.198 0.131 0.123 0.168 0.019 0.026 0.02 0.02 0.022 0.02 0.020 0.019 0.024 0.02 0.019 0.026 0.025 0.03 0.78
B 0.20 1.000 0.005 0.097 0.012 0.005 0.005 0.01 0.01 0.008 0.01 0.013 0.009 0.016 0.01 0.003 0.017 0.008 0.01 0.08
C 0.13 0.005 1.000 0.013 0.093 0.018 0.020 0.01 0.02 0.022 0.02 0.021 0.013 0.030 0.02 0.013 0.018 0.016 0.02 0.07
D 0.12 0.097 0.013 1.000 0.008 0.048 0.040 0.05 0.05 0.054 0.04 0.053 0.048 0.052 0.04 0.055 0.047 0.044 0.05 0.06
E 0.17 0.012 0.093 0.008 1.000 0.003 0.009 0.01 0.01 0.023 0.02 0.003 0.001 0.008 0.02 0.003 0.008 0.006 0.02 0.06
F 0.02 0.005 0.018 0.048 0.003 1.000 0.045 0.03 0.05 0.036 0.02 0.043 0.028 0.030 0.04 0.035 0.036 0.042 0.04 0.05
G 0.03 0.005 0.020 0.040 0.009 0.045 1.000 0.04 0.05 0.034 0.02 0.035 0.029 0.047 0.03 0.055 0.042 0.041 0.05 0.05
H 0.02 0.015 0.012 0.046 0.011 0.029 0.040 1.00 0.04 0.033 0.04 0.041 0.043 0.032 0.04 0.043 0.051 0.038 0.05 0.04
I 0.02 0.010 0.023 0.047 0.010 0.054 0.045 0.04 1.00 0.042 0.02 0.032 0.051 0.045 0.04 0.035 0.028 0.036 0.04 0.04
J 0.02 0.008 0.022 0.054 0.023 0.036 0.034 0.03 0.04 1.000 0.03 0.035 0.026 0.020 0.03 0.047 0.048 0.036 0.03 0.03
K 0.02 0.010 0.016 0.035 0.025 0.019 0.023 0.04 0.02 0.025 1.00 0.030 0.022 0.028 0.03 0.021 0.024 0.029 0.03 0.03
L 0.02 0.013 0.021 0.053 0.003 0.043 0.035 0.04 0.03 0.035 0.03 1.000 0.020 0.049 0.04 0.028 0.032 0.030 0.04 0.04
M 0.02 0.009 0.013 0.048 0.001 0.028 0.029 0.04 0.05 0.026 0.02 0.020 1.000 0.048 0.04 0.043 0.042 0.032 0.04 0.05
N 0.02 0.016 0.030 0.052 0.008 0.030 0.047 0.03 0.04 0.020 0.03 0.049 0.048 1.000 0.03 0.046 0.047 0.052 0.04 0.04
O 0.02 0.011 0.015 0.043 0.016 0.035 0.032 0.04 0.04 0.032 0.03 0.042 0.037 0.026 1.00 0.033 0.050 0.049 0.03 0.03
P 0.02 0.003 0.013 0.055 0.003 0.035 0.055 0.04 0.03 0.047 0.02 0.028 0.043 0.046 0.03 1.000 0.035 0.032 0.03 0.03
Q 0.03 0.017 0.018 0.047 0.008 0.036 0.042 0.05 0.03 0.048 0.02 0.032 0.042 0.047 0.05 0.035 1.000 0.032 0.04 0.03
R 0.02 0.008 0.016 0.044 0.006 0.042 0.041 0.04 0.04 0.036 0.03 0.030 0.032 0.052 0.05 0.032 0.032 1.000 0.04 0.05
S 0.03 0.013 0.018 0.052 0.016 0.036 0.046 0.05 0.04 0.029 0.03 0.042 0.041 0.037 0.03 0.031 0.041 0.044 1.00 0.04
T 0.78 0.082 0.065 0.063 0.064 0.054 0.045 0.04 0.04 0.028 0.03 0.035 0.046 0.042 0.03 0.034 0.030 0.053 0.04 1.00
```

Predictor Transformations (e.g., PCA)

Modeling

Feature Selection

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

```
# Convert factors to dummies (retaining non-factors and also keeping the target as a factor)
dummies <- dummyVars(Disorder_Subclass ~. , data = train_df[ , sapply(train_df, is.factor)])
train_df <- cbind(Disorder_Subclass = train_df$Disorder_Subclass, train_df[ , !sapply(train_df, is.factor)], data.frame(p
redict(dummies, newdata = train_df)))
dummies <- dummyVars(Disorder_Subclass ~. , data = test_df[ , sapply(test_df, is.factor)])
test_df <- cbind(Disorder_Subclass = test_df$Disorder_Subclass, test_df[ , !sapply(test_df, is.factor)], data.frame(predi
ct(dummies, newdata = test_df)))

# Create Random Forest weight vector based on class priors
priors <- as.list(prop.table(table(train_df$Disorder_Subclass)))
wts <- data.frame(Disorder_Subclass = train_df$Disorder_Subclass, w = 0.0)
for (n in 1:length(priors))
  wts[wts$Disorder_Subclass == names(priors[n]), ]$w <- priors[[n]]

# Train the model (using defaults)
rf_fit <- randomForest(x = train_df,
                       y = train_df$Disorder_Subclass,
                       xtest = test_df,
                       ytest = test_df$Disorder_Subclass,
                       weights = as.vector(wts$w),
                       importance = TRUE)

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

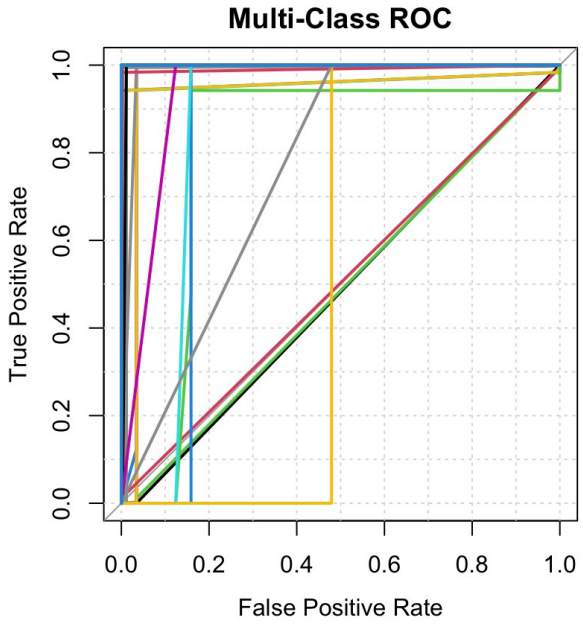
```
Call:
 randomForest(x = train_df, y = train_df$Disorder_Subclass, xtest = test_df,      ytest = test_df$Disorder_Subclass, weig
hts = as.vector(wts$w),      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 8

        OOB estimate of  error rate: 4%
Confusion matrix:
                                  Class A Class B Class C Class D Class E Class F Class G Class H Class I class.erro
r
Alzheimers (A)                          0       0       4     112       0       0       0       3       0       1.00
```

```
0
Cancer (B)                                0       0       0      69       0       0       0       0       5        1.00
0
Cystic.fibrosis (C)                       0       0    2619       0       0       0       0       3       0        0.00
1
Diabetes (D)                              0       0       5    1359       0       0       0      27       4        0.02
6
Hemochromatosis (E)                       0       0       2       0     853       0       0      41     137        0.17
4
Lebers.hereditary.optic.neuropathy (F)    0       0      16      17       0     270       0     178       5        0.44
4
Leigh.syndrome (G)                        0       0       0       0       0       0    3915       0       0        0.00
0
Mitochondrial.myopathy (H)                0       0       0       0       0       0       0    3362       0        0.00
0
Tay.Sachs (I)                             0       0       0       0       0       0       0       7    2145        0.00
3
                Test set error rate: 4%
Confusion matrix:
                                      Class A Class B Class C Class D Class E Class F Class G Class H Class I class.erro
r
Alzheimers (A)                            0       0       0      28       0       0       0       0       1        1.00
0
Cancer (B)                                0       0       0      18       0       0       0       0       0        1.00
0
Cystic.fibrosis (C)                       0       0     654       0       0       0       0       0       1        0.00
2
Diabetes (D)                              0       0       2     342       0       0       0       3       1        0.01
7
Hemochromatosis (E)                       0       0       0       0     217       0       0       9      32        0.15
9
Lebers.hereditary.optic.neuropathy (F)    0       0       2       5       0      56       0      58       0        0.53
7
Leigh.syndrome (G)                        0       0       0       0       0       0     978       0       0        0.00
0
Mitochondrial.myopathy (H)                0       0       0       0       0       0       0     840       0        0.00
0
Tay.Sachs (I)                             0       0       0       0       0       0       0       0     538        0.00
0
```

```
# Genrate ROC for test results
rf_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
                         predictor = as.numeric(rf_fit$test$predicted))
auc(rf_roc)
```

```
Multi-class area under the curve: 0.9
```

```
# Plot mulit-class ROC
rocs <- rf_roc[['rocs']]
par(pty = "s")
plot.roc(rocs[[1]], grid = TRUE, legacy.axes = TRUE,
         main = "Multi-Class ROC", xlab = "False Positive Rate", ylab = "True Positive Rate")
lines <- sapply(2:length(rocs), function(x) lines.roc(rocs[[x]], col = x))
```



```
dev <- dev.off()
```

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↩