Predicting Genetic Disorders

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

06/27/2022

```
## Warning: package 'caret' was built under R version 4.0.5
```

Data Load and Validation

```
gd_df = read.csv(file="/Users/sakshyamdahal/Desktop/MS_Data_Science/Applied Predictive
        Modeling/Final_project/train_genetic_disorders.csv", header= TRUE)
dim(gd_df)
```

```
[1] 22083    45
```

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 Hospital For Women" "" "" ...
 $ Location.of.Institute                           : chr  "55 FRUIT ST\nCENTRAL, MA
02114\n(42.36247485742686, -71.06924724545246)" "1515 COMMONWEALTH AV\nALLSTON/BRIGHTON,
MA 02135\n(42.34665771451756, -71.14136122385321)" "-" "55 FRUIT ST\nCENTRAL, 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" "" "Multifactorial genetic inheritance disorders" "Mitochondrial
genetic inheritance disorders" ...
 $ Disorder.Subclass                               : chr  "Leber's hereditary optic
neuropathy" "Cystic fibrosis" "Diabetes" "Leigh syndrome" ...
```

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

Initial feature reduction

Uninformative Feature Reduced

```
# 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)]

dim(gd_df)
```

```
[1] 22083    31
```

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("Deleted rows with missing labels: ", format(nrow(gd_hold_df), format = "d", big.mark
        = ","), sep = "")
```

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

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

```
 Remaining 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 Partition

Split the dataframe as per target class (Disorder.Subclass vector)

There is certain class imbalance problem. We will split the data frame using createDataPartition function using target class(Disorder.Subclass) which helps us to get even balance of all classes in both train and test data

```
# 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
```

Handling Missing Values

Checking Mislabeled Data and Missing Values

```
library(dplyr)
library(questionr)

#lapply(train_df, unique)

# changing values which are labeled as not available and no record to a readable format
        'NA'
train_df[ train_df == "Not available"] <- NA
test_df[ test_df == "Not available"] <- NA

train_df[train_df == "No record"] <- NA
test_df[test_df == "No record"] <- NA

#changing None to "No" (there is None in one vector
        "Autopsy.shows.birth.defect..if.applicable." where none birth defects resemble no
        class)
train_df[train_df == "None"] <- "No"
test_df[test_df == "None"] <- "No"


freq.na(train_df)
```

	missing	%
Birth.asphyxia	6908	46
Mother.s.age	3947	26
Father.s.age	3853	25
Symptom.2	1457	10
Symptom.5	1451	10
White.Blood.cell.count..thousand.per.microliter.	1437	9
No..of.previous.abortion	1418	9
Symptom.1	1410	9
Symptom.4	1382	9
Symptom.3	1363	9
Patient.Age	960	6
Disorder.Subclass	0	0
Genes.in.mother.s.side	0	0
Inherited.from.father	0	0
Maternal.gene	0	0
Paternal.gene	0	0
Blood.cell.count..mcL.	0	0
Respiratory.Rate..breaths.min.	0	0
Heart.Rate..rates.min	0	0
Follow.up	0	0
Gender	0	0
Autopsy.shows.birth.defect..if.applicable.	0	0
Folic.acid.details..peri.conceptional.	0	0

```
H.O.serious.maternal.illness                          0  0
H.O.radiation.exposure..x.ray.                        0  0
H.O.substance.abuse                                   0  0
Assisted.conception.IVF.ART                           0  0
History.of.anomalies.in.previous.pregnancies          0  0
Birth.defects                                         0  0
Blood.test.result                                     0  0
Genetic.Disorder                                      0  0
```

```
sum(is.na(train_df))
```

```
[1] 25586
```

Note:We will be removing “Birth.asphyxia” vector from our data as there is 46% missing data. Here, the imputation might create bias model.Further,we will be performing median imputation for integer vectors. For categorical vectors we will be encoding the missing value as “not provided”. Here we can also do mode imputation but we would like to learn if the missing value has any relation with target. For numeric vectors we will be applying mean imputation.

```
train_df <- subset(train_df, select = -c(Birth.asphyxia))
test_df <- subset(test_df, select = -c(Birth.asphyxia))

# 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",
                    "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[cols_tofill][train_df[cols_tofill] == ""] <- "notprovided"
test_df[cols_tofill][test_df[cols_tofill] == ""] <- "notprovided"

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)


#lapply(train_df, unique)
```

Pre-processing

Feature Pre-processing (including variable types/formats, names)

```
# preprocess variables
```

```
factor_cols <- c("Genes.in.mother.s.side",
                 "Inherited.from.father",
                 "Maternal.gene",
                 "Paternal.gene",
                 "Respiratory.Rate..breaths.min.",
                 "Heart.Rate..rates.min",
                 "Follow.up",
                 "Gender",
                 "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",
                 "Genetic.Disorder")
train_df[factor_cols] <- lapply(train_df[factor_cols], factor)
test_df[factor_cols] <- lapply(test_df[factor_cols], factor)
# Note factors can be changed in dummy variables for better performances while in
        modeling.

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

```
'data.frame':   15158 obs. of  30 variables:
 $ Disorder.Subclass                             : Factor w/ 9 levels
"Alzheimers","Cancer",..: 6 3 4 7 2 9 9 7 4 3 ...
 $ Patient.Age                                   : int  2 4 6 12 11 3 3 11 4 7 ...
 $ Genes.in.mother.s.side                        : Factor w/ 2 levels "No","Yes": 2 2 2
2 2 2 1 1 1 1 ...
 $ Inherited.from.father                         : Factor w/ 3 levels
"No","notprovided",..: 1 3 1 1 1 1 1 1 3 1 ...
 $ Maternal.gene                                 : Factor w/ 3 levels
"No","notprovided",..: 3 1 1 3 2 3 3 3 3 1 ...
 $ Paternal.gene                                 : Factor w/ 2 levels "No","Yes": 1 1 1
1 2 2 2 1 2 2 ...
 $ Blood.cell.count..mcL.                        : num  4.76 4.91 4.89 4.71 4.72 ...
 $ Mother.s.age                                  : int  35 35 41 21 32 35 40 45 44 35
...
 $ Father.s.age                                  : int  42 23 22 42 42 63 42 44 42 42
...
 $ Respiratory.Rate..breaths.min.                : Factor w/ 3 levels "Normal (30-
60)",..: 1 3 1 3 3 1 3 3 3 1 ...
 $ Heart.Rate..rates.min                         : Factor w/ 3 levels
```

```
"Normal","notprovided",..: 1 1 3 1 3 2 1 3 3 3 ...
 $ Follow.up                                      : Factor w/ 3 levels
"High","Low","notprovided": 1 1 2 1 2 2 2 2 2 2 ...
 $ Gender                                         : Factor w/ 4 levels
"Ambiguous","Female",..: 4 4 4 3 3 3 4 3 3 3 ...
 $ Autopsy.shows.birth.defect..if.applicable.     : Factor w/ 4 levels "No","Not
applicable",..: 2 1 2 1 2 2 2 2 2 1 ...
 $ Folic.acid.details..peri.conceptional.         : Factor w/ 3 levels
"No","notprovided",..: 1 3 3 1 1 2 3 3 3 3 ...
 $ H.O.serious.maternal.illness                   : Factor w/ 3 levels
"No","notprovided",..: 2 3 1 3 3 3 3 3 1 3 ...
 $ H.O.radiation.exposure..x.ray.                 : Factor w/ 4 levels "No","Not
applicable",..: 1 2 4 3 3 1 1 1 1 3 ...
 $ H.O.substance.abuse                            : Factor w/ 4 levels "No","Not
applicable",..: 1 2 3 2 2 2 3 1 1 3 ...
 $ Assisted.conception.IVF.ART                    : Factor w/ 3 levels
"No","notprovided",..: 1 1 3 2 3 3 1 1 3 1 ...
 $ History.of.anomalies.in.previous.pregnancies   : Factor w/ 3 levels
"No","notprovided",..: 3 3 3 3 1 1 3 3 3 3 ...
 $ No..of.previous.abortion                       : int  2 2 4 1 4 3 1 0 1 2 ...
 $ Birth.defects                                  : Factor w/ 3 levels
"Multiple","notprovided",..: 2 1 3 3 1 1 3 1 1 1 ...
 $ White.Blood.cell.count..thousand.per.microliter.: num  9.86 5.52 7.46 7.92 4.1 ...
 $ Blood.test.result                              : Factor w/ 5 levels
"abnormal","inconclusive",..: 4 3 3 2 4 3 2 5 1 5 ...
 $ Symptom.1                                      : int  1 1 0 0 0 0 0 1 0 0 ...
 $ Symptom.2                                      : int  1 -1 1 0 0 0 0 1 0 1 ...
 $ Symptom.3                                      : int  1 1 1 1 0 0 1 1 1 1 ...
 $ Symptom.4                                      : int  1 1 1 0 0 0 -1 0 1 1 ...
 $ Symptom.5                                      : int  1 0 1 0 -1 0 0 1 1 1 ...
 $ Genetic.Disorder                               : Factor w/ 4 levels "Mitochondrial
genetic inheritance disorders",..: 1 3 2 1 2 4 4 1 2 4 ...
```

Collinearity and Dependencies

```r
# 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.mother.s.side (B)
Inherited.from.father (C)
Maternal.gene (D)
Paternal.gene (E)
Respiratory.Rate..breaths.min. (F)
Heart.Rate..rates.min (G)
Follow.up (H)
Gender (I)
Autopsy.shows.birth.defect..if.applicable. (J)
Folic.acid.details..peri.conceptional. (K)
H.O.serious.maternal.illness (L)
H.O.radiation.exposure..x.ray. (M)
H.O.substance.abuse (N)
Assisted.conception.IVF.ART (O)
History.of.anomalies.in.previous.pregnancies (P)
Birth.defects (Q)
Blood.test.result (R)
Genetic.Disorder (S)
```

```
cscorr
```

```
     A     B     C     D     E     F     G    H    I     J     K     L     M     N     O
P     Q    R    S
A 1.00 0.198 0.131 0.123 0.168 0.019 0.026 0.02 0.02 0.018 0.020 0.019 0.022 0.018 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 0.009 0.013 0.009 0.015 0.008 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 0.015 0.021 0.013 0.013 0.009 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 0.035 0.053 0.048 0.034 0.026 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 0.025 0.003 0.001 0.008 0.009 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 0.018 0.043 0.028 0.023 0.012 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 0.023 0.035 0.029 0.025 0.017 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 0.038 0.041 0.043 0.019 0.024 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 0.022 0.032 0.051 0.024 0.022 0.035
0.028 0.036 0.04 0.04
J 0.02 0.009 0.015 0.035 0.025 0.018 0.023 0.04 0.02 1.000 0.028 0.019 0.015 0.016 0.016
0.024 0.029 0.03 0.03
K 0.02 0.013 0.021 0.053 0.003 0.043 0.035 0.04 0.03 0.028 1.000 0.020 0.012 0.024 0.028
0.032 0.030 0.04 0.04
L 0.02 0.009 0.013 0.048 0.001 0.028 0.029 0.04 0.05 0.019 0.020 1.000 0.020 0.022 0.043
0.042 0.032 0.04 0.05
M 0.02 0.015 0.013 0.034 0.008 0.023 0.025 0.02 0.02 0.015 0.012 0.020 1.000 0.015 0.024
0.031 0.030 0.03 0.02
N 0.02 0.008 0.009 0.026 0.009 0.012 0.017 0.02 0.02 0.016 0.024 0.022 0.015 1.000 0.017
0.027 0.019 0.02 0.02
O 0.02 0.003 0.013 0.055 0.003 0.035 0.055 0.04 0.03 0.016 0.028 0.043 0.024 0.017 1.000
0.035 0.032 0.03 0.03
P 0.03 0.017 0.018 0.047 0.008 0.036 0.042 0.05 0.03 0.024 0.032 0.042 0.031 0.027 0.035
1.000 0.032 0.04 0.03
Q 0.02 0.008 0.016 0.044 0.006 0.042 0.041 0.04 0.04 0.029 0.030 0.032 0.030 0.019 0.032
0.032 1.000 0.04 0.05
R 0.03 0.013 0.018 0.052 0.016 0.036 0.046 0.05 0.04 0.028 0.042 0.041 0.028 0.024 0.031
0.041 0.044 1.00 0.04
S 0.78 0.082 0.065 0.063 0.064 0.054 0.045 0.04 0.04 0.030 0.035 0.046 0.024 0.023 0.034
0.030 0.053 0.04 1.00
```

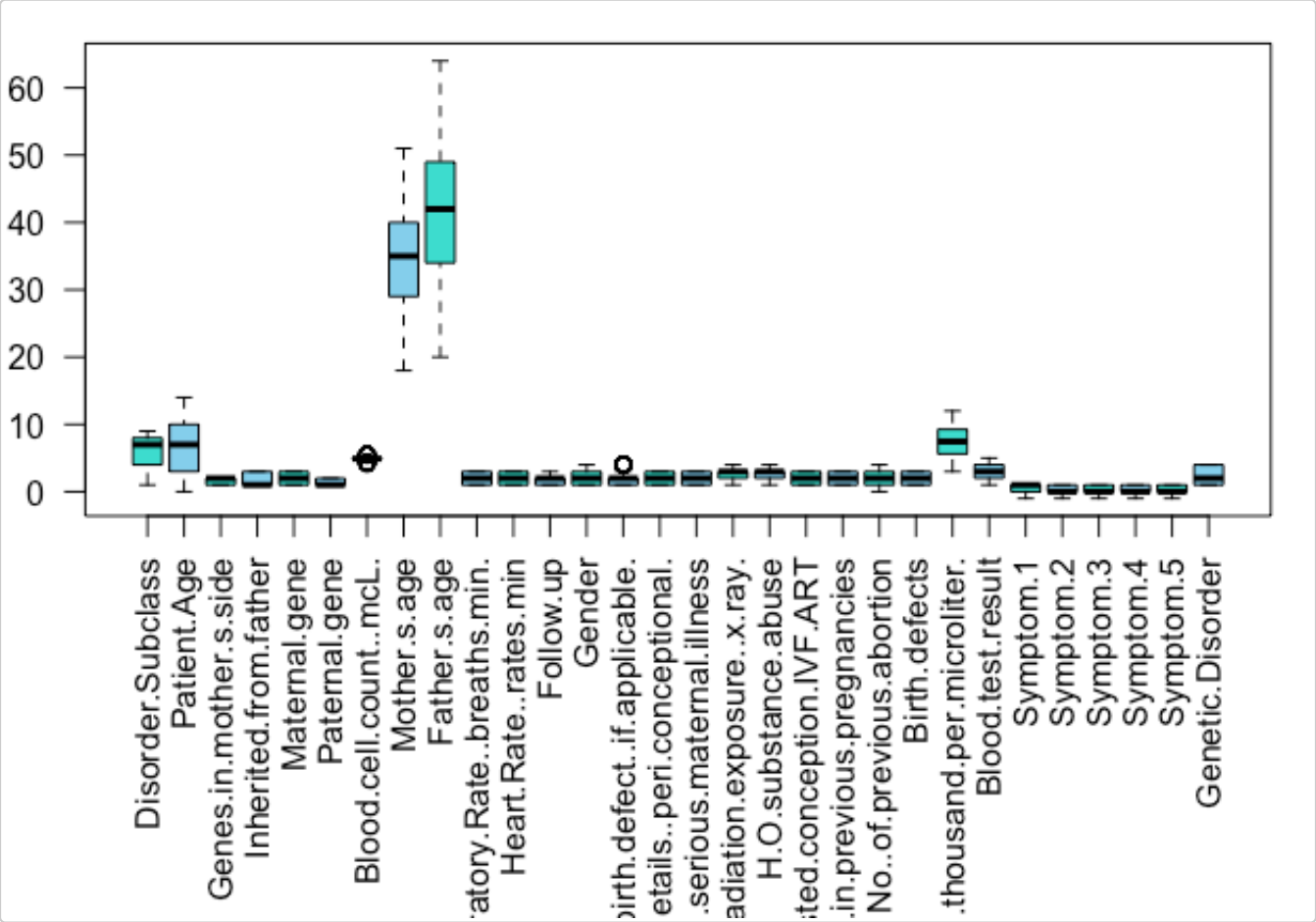
Exploratory Data Analysis(EDA)

```
dim(train_df)
```

```
[1] 15158    30
```

Outlier detection

```
par(mar=c(10,2,1,1))
boxplot(train_df, las=2, col = c("turquoise","skyblue"))
```

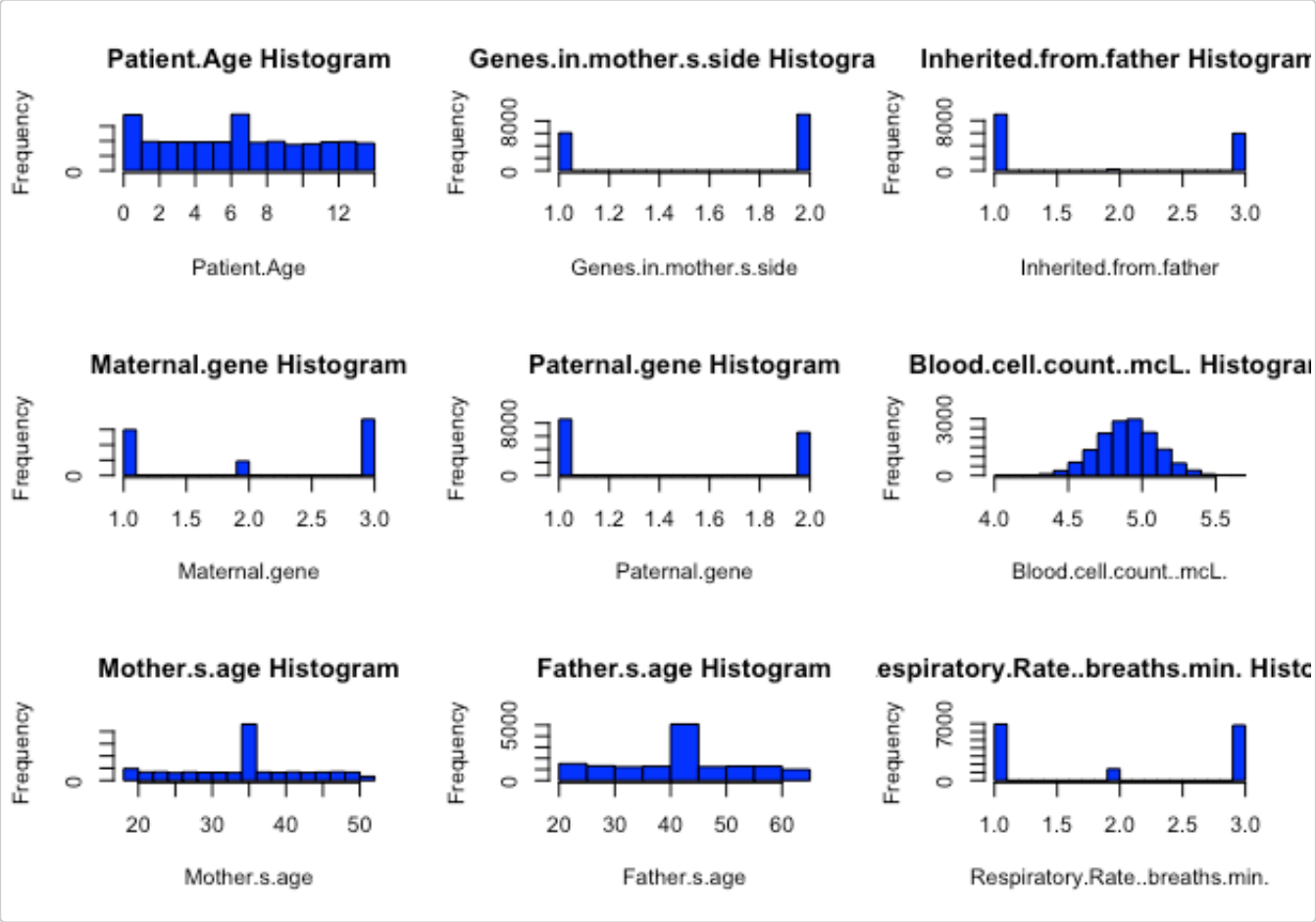


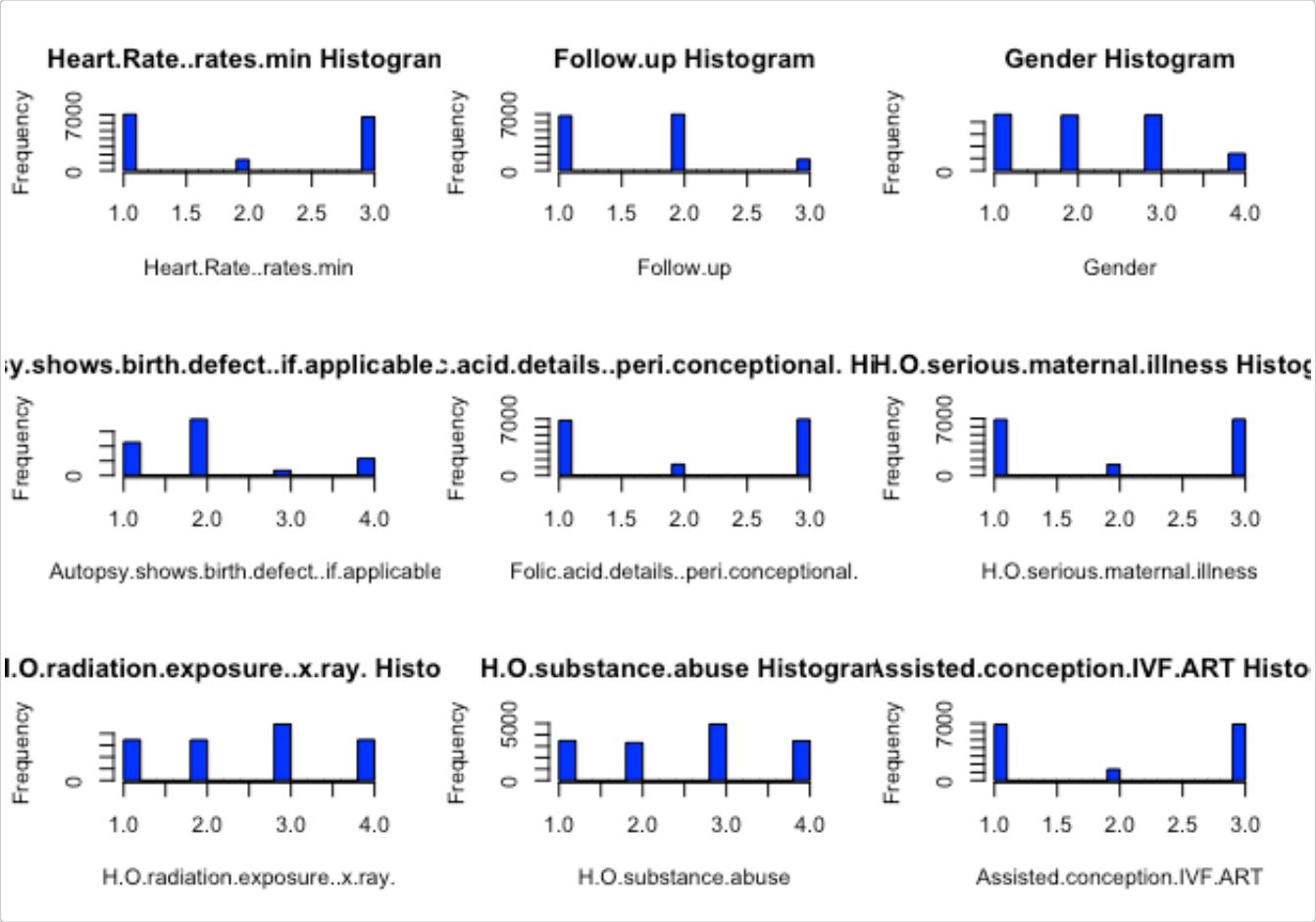
Note: Our data looks good on as there seems to be no outliers in the data.

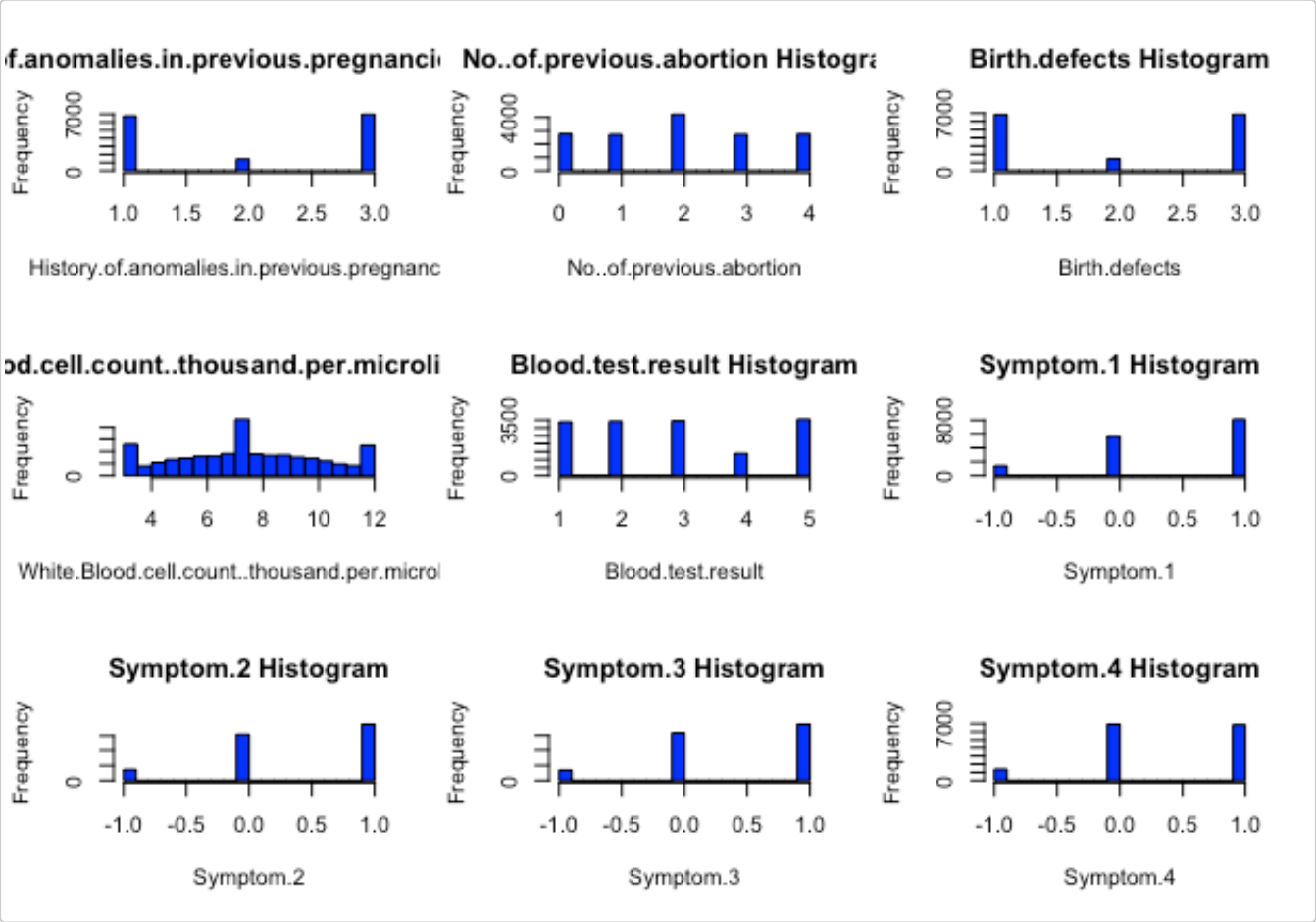
Frequency distribution

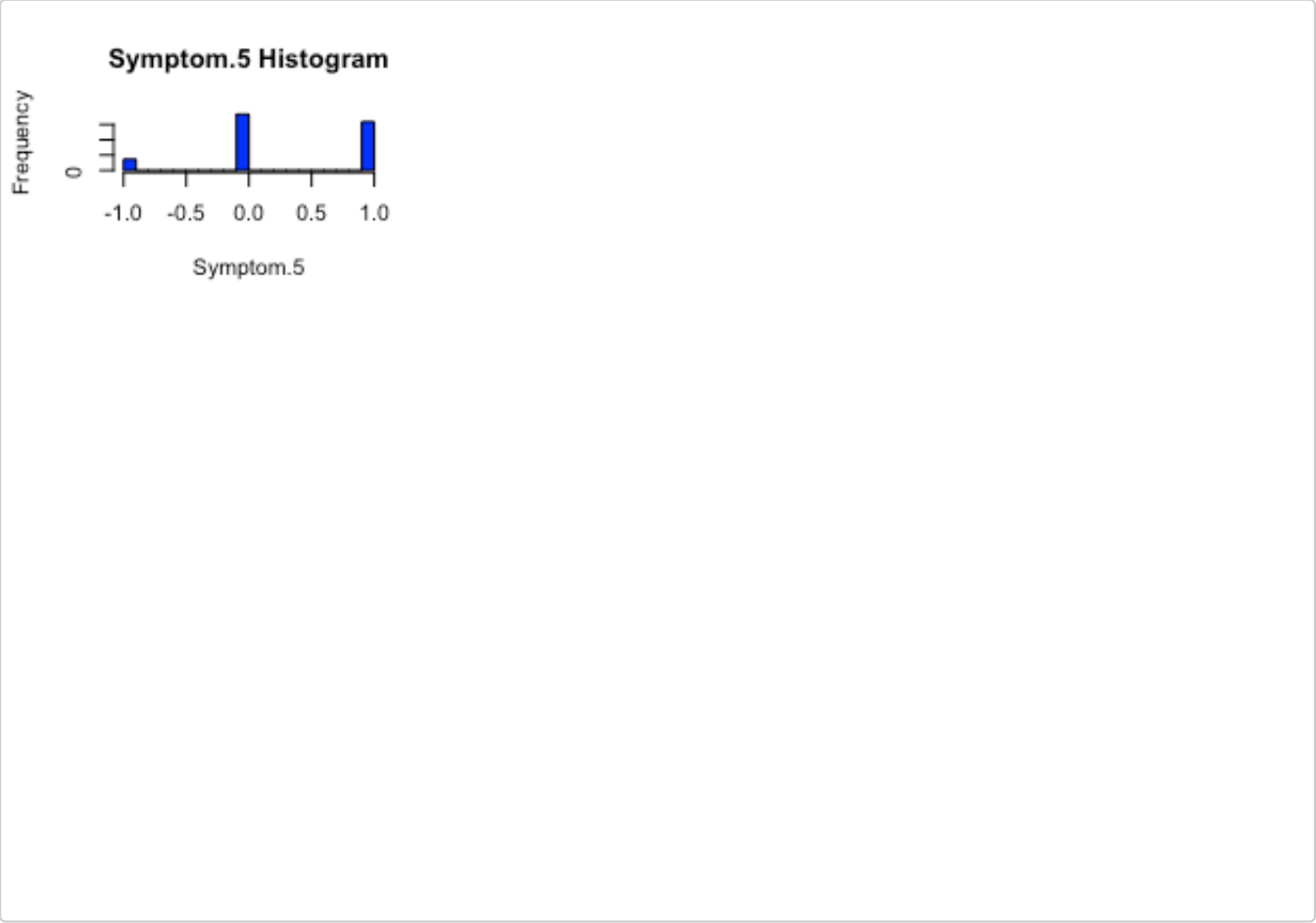
```
pred_for_hist <- train_df[,2:29]
pred_for_hist <- pred_for_hist %>% mutate_if(is.character, as.numeric)
pred_for_hist <- pred_for_hist %>% mutate_if(is.factor, as.numeric)
par(mfrow = c(3, 3))

for (i in 1:ncol(pred_for_hist)) {
  hist(pred_for_hist[ ,i], xlab = names(pred_for_hist[i]), main =
         paste(names(pred_for_hist[i]), "Histogram"), col="blue")
}
```


Patient.Age Histogram
Frequency
0
0 2 4 6 8 12
Patient.Age
Genes.in.mother.s.side Histogra
Frequency
0 8000
1.0 1.2 1.4 1.6 1.8 2.0
Genes.in.mother.s.side
Inherited.from.father Histogram
Frequency
0 8000
1.0 1.5 2.0 2.5 3.0
Inherited.from.father
Maternal.gene Histogram
Frequency
0
1.0 1.5 2.0 2.5 3.0
Maternal.gene
Paternal.gene Histogram
Frequency
0 8000
1.0 1.2 1.4 1.6 1.8 2.0
Paternal.gene
Blood.cell.count..mcL. Histogra
Frequency
0 3000
4.0 4.5 5.0 5.5
Blood.cell.count..mcL.
Mother.s.age Histogram
Frequency
0
20 30 40 50
Mother.s.age
Father.s.age Histogram
Frequency
0 5000
20 30 40 50 60
Father.s.age
espiratory.Rate..breaths.min. Histo
Frequency
0 7000
1.0 1.5 2.0 2.5 3.0
Respiratory.Rate..breaths.min.


Heart.Rate..rates.min Histogran
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
Heart.Rate..rates.min
Follow.up Histogram
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
Follow.up
Gender Histogram
Frequency
0
1.0
2.0
3.0
4.0
Gender
y.shows.birth.defect..if.applicable
Frequency
0
1.0
2.0
3.0
4.0
Autopsy.shows.birth.defect..if.applicable
.acid.details..peri.conceptional. Hi
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
Folic.acid.details..peri.conceptional.
H.O.serious.maternal.illness Histog
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
H.O.serious.maternal.illness
I.O.radiation.exposure..x.ray. Histo
Frequency
0
1.0
2.0
3.0
4.0
H.O.radiation.exposure..x.ray.
H.O.substance.abuse Histogran
Frequency
0
5000
1.0
2.0
3.0
4.0
H.O.substance.abuse
Assisted.conception.IVF.ART Histo
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
Assisted.conception.IVF.ART


f.anomalies.in.previous.pregnanci
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
History.of.anomalies.in.previous.pregnanc
No..of.previous.abortion Histogra
Frequency
0
4000
0
1
2
3
4
No..of.previous.abortion
Birth.defects Histogram
Frequency
0
7000
1.0
1.5
2.0
2.5
3.0
Birth.defects
od.cell.count..thousand.per.microli
Frequency
0
4
6
8
10
12
White.Blood.cell.count..thousand.per.micro
Blood.test.result Histogram
Frequency
0
3500
1
2
3
4
5
Blood.test.result
Symptom.1 Histogram
Frequency
0
8000
-1.0
-0.5
0.0
0.5
1.0
Symptom.1
Symptom.2 Histogram
Frequency
0
-1.0
-0.5
0.0
0.5
1.0
Symptom.2
Symptom.3 Histogram
Frequency
0
-1.0
-0.5
0.0
0.5
1.0
Symptom.3
Symptom.4 Histogram
Frequency
0
7000
-1.0
-0.5
0.0
0.5
1.0
Symptom.4




```
#par(mfrow = c(4, 4))
#for (i in 1:ncol(pred_for_hist)) {
#d <- density(pred_for_hist[,i], na.rm = TRUE)
#plot(d, main = paste(names(pred_for_hist[i]), "Density"))
#polygon(d, col="blue")
#}
```

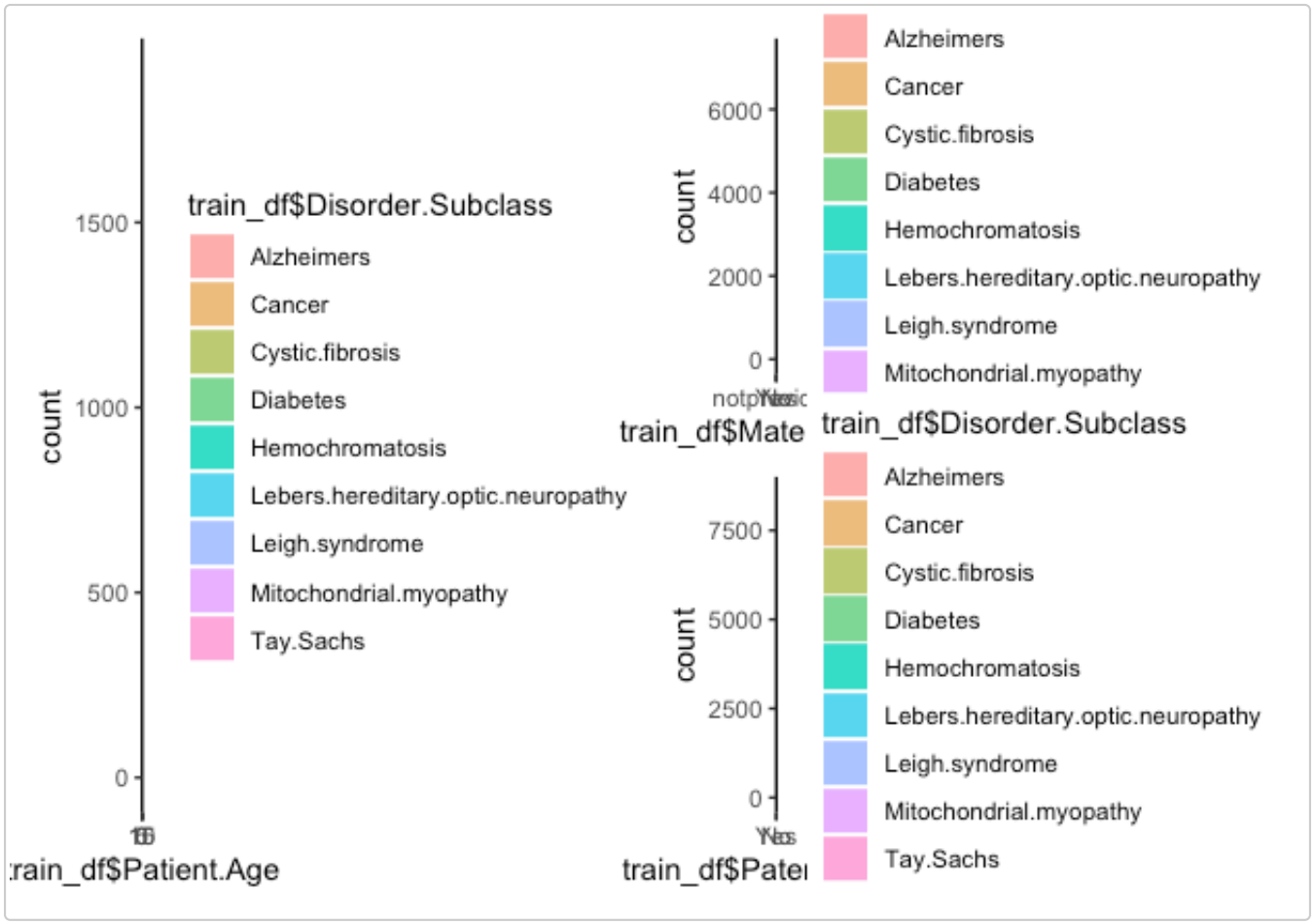
Relation with target based on our hypothesis that the maternal and paternal genes might be cause of transmission of the genetic diorders.

```
library(ggplot2)
#install.packages("patchwork")
#install.packages("cowplot")

library(patchwork)
library(cowplot)
p1 <- ggplot(train_df, aes(x = train_df$Patient.Age, fill = train_df$Disorder.Subclass))
        +geom_bar() + theme_classic()+scale_fill_hue(c=60, l=80)
```

```
p2 <- ggplot(train_df, aes(x = train_df$Maternal.gene, fill =
        train_df$Disorder.Subclass)) +geom_bar() +theme_classic()+ scale_fill_hue(c=60,
        l=80)
p3 <- ggplot(train_df, aes(x = train_df$Paternal.gene, fill =
        train_df$Disorder.Subclass)) +geom_bar() +theme_classic()+ scale_fill_hue(c=60,
        l=80)

p1+p2/p3
```



Modeling

Assigning target and Predictors

```
#test train before dummy
train_x <- train_df[,2:29]
train_y <- train_df[,1]

test_x <- test_df[,2:29]
test_y <- test_df[,1]
```

```
dim(train_x)
```

```
[1] 15158    28
```

```
dim(test_x)
```

```
[1] 3785   28
```

```
#subsetting numeric columns and character for changing categorical into dummy
library(dplyr)
# Subset numeric columns with dplyr

numeric_pred_train <- select_if(train_x, is.numeric)
numeric_pred_test<- select_if(test_x, is.numeric)

# Subset categorical columns with dplyr
cat_pred_train <- select_if(train_x,is.factor)
cat_pred_test <- select_if(test_x,is.factor)

dim(numeric_pred_train)
```

```
[1] 15158    11
```

```
dim(numeric_pred_test)
```

```
[1] 3785   11
```

```
dim(cat_pred_train)
```

```
[1] 15158    17
```

```
dim(cat_pred_test)
```

```
[1] 3785   17
```

```
#encode to dummy
library(lattice)
dummies <- dummyVars(~ ., data=cat_pred_train[,1:17])
dummy_cat_df <- predict(dummies, cat_pred_train[,1:17])

dummies_test <- dummyVars(~ ., data=cat_pred_test[,1:17])
dummy_cat_df_test<- predict(dummies, cat_pred_test[,1:17])

#ready to model train test
train_x<- as.data.frame(cbind(numeric_pred_train, dummy_cat_df))
train_y <- as.factor(train_y)

test_x<- as.data.frame(cbind(numeric_pred_test, dummy_cat_df_test))
test_y <- as.factor(test_y)
```

Setting control function for our multiclass classification

```
ctrl <- trainControl(method = "cv",
                     summaryFunction = multiClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)


set.seed(476)

#nnetGrid <- expand.grid(size=1:3, decay=c(0,0.1,0.2,0.3,0.4,0.5,1,2))

#nnetFit <- train(x =  train_x,
               # y = train_y,
              #  method = "nnet",
              #  tuneGrid = nnetGrid,
              #  metric = "ROC",
             #   trace = FALSE,
              #  maxit = 2000,
               # trControl = ctrl)
```

SVM (Support Vector machine)

```
#sigmaEst <- kernlab::sigest(as.matrix(train_x))
#svmgrid <- expand.grid(sigma = sigmaEst, C = 2^seq(-4,+4))
```

```
#set.seed(476)
#svmRFit <- train(x = train_x ,
                  # y = train_y,
                  # method = "svmRadial",
                  # tuneGrid = svmgrid,
                  # metric = "ROC",
                   #trControl = ctrl)
```

---

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