

Predicting Genetic Disorders

Emma Oo¹, Sindhu Bhattarai², Dave Friesen³

06/27/2022

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 Hospi
tal For Women" "" "" ...
 $ Location.of.Institute : chr "55 FRUIT ST\nCENTRAL, MA 02114\n(42.36247485742686, -71.069247245
45246)" "1515 COMMONWEALTH AV\nALLSTON/BRIGHTON, MA 02135\n(42.34665771451756, -71.14136122385321)" "-" "55 FRUIT ST\nCENTRA
L, 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.birthing.defect..if.applicable. : chr "Not applicable" "None" "Not applicable" "No" ...
 $ Place.of.birthing : 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 g
enetic inheritance disorders" "Mitochondrial genetic inheritance disorders" ...
 $ Disorder.Subclass : chr "Leber's hereditary optic neuropathy" "Cystic fibrosis" "Diabetes"
"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",
              "Birth.asphyxia",
              "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("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

# 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 Partitioning

```
# 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	Blank	Z	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								
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

```

# 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")

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 should 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",
  "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_mothers_side",
  "Genes_fathers_side",
  "Maternal_gene",
  "Paternal_gene",
  "Blood_cell_count",
  "Mothers_age",
  "Fathers_age",
  "Respiratory_Rate",
  "Heart_Rate",
  "Follow_up",
  "Gender",
  "Autopsy_birth_defect",
  "Folic_acid_conceptional",
  "HO_maternal_illness",
  "HO_radiation_exposure",
  "HO_substance_abuse",
  "Assisted_conception",
  "Previous_pregnancies_issues",
  "Previous_abortions",
  "Birth_defects",
  "White_Blood_cell_count",
  "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

```

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)

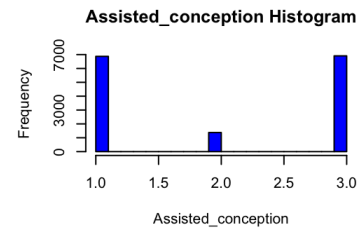
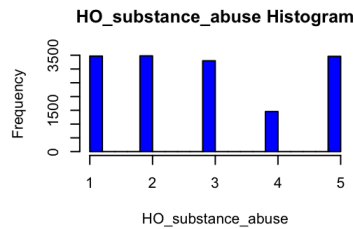
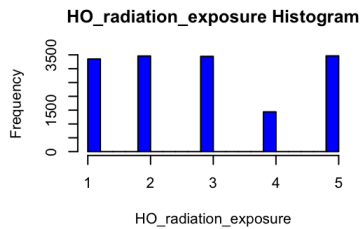
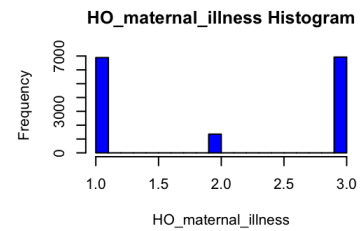
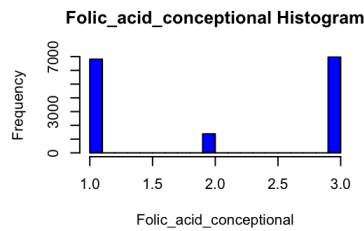
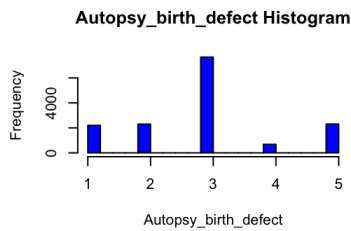
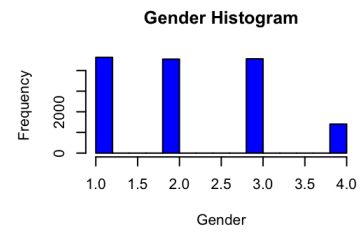
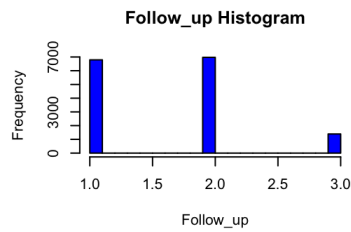
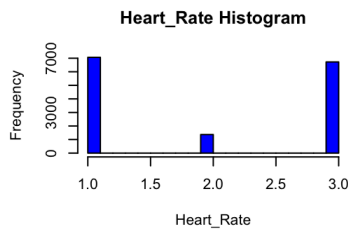
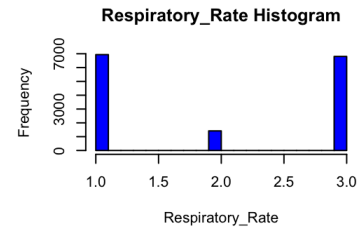
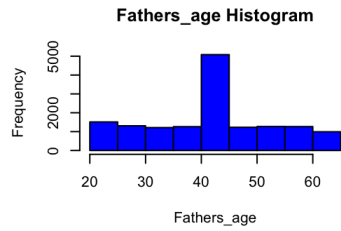
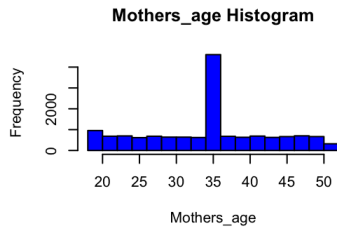
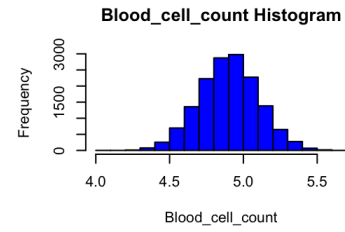
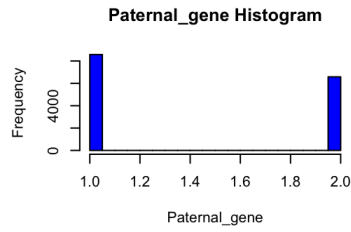
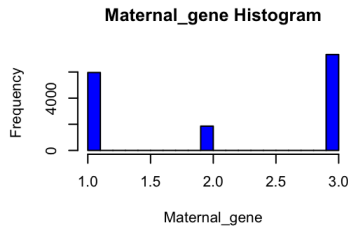
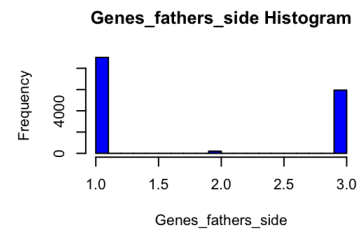
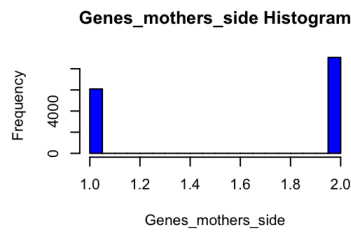
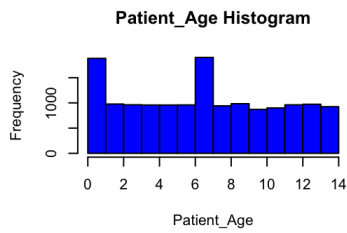
```
# Generate updated summary of base dataset which includes these characteristics
univariate(train_df)
```

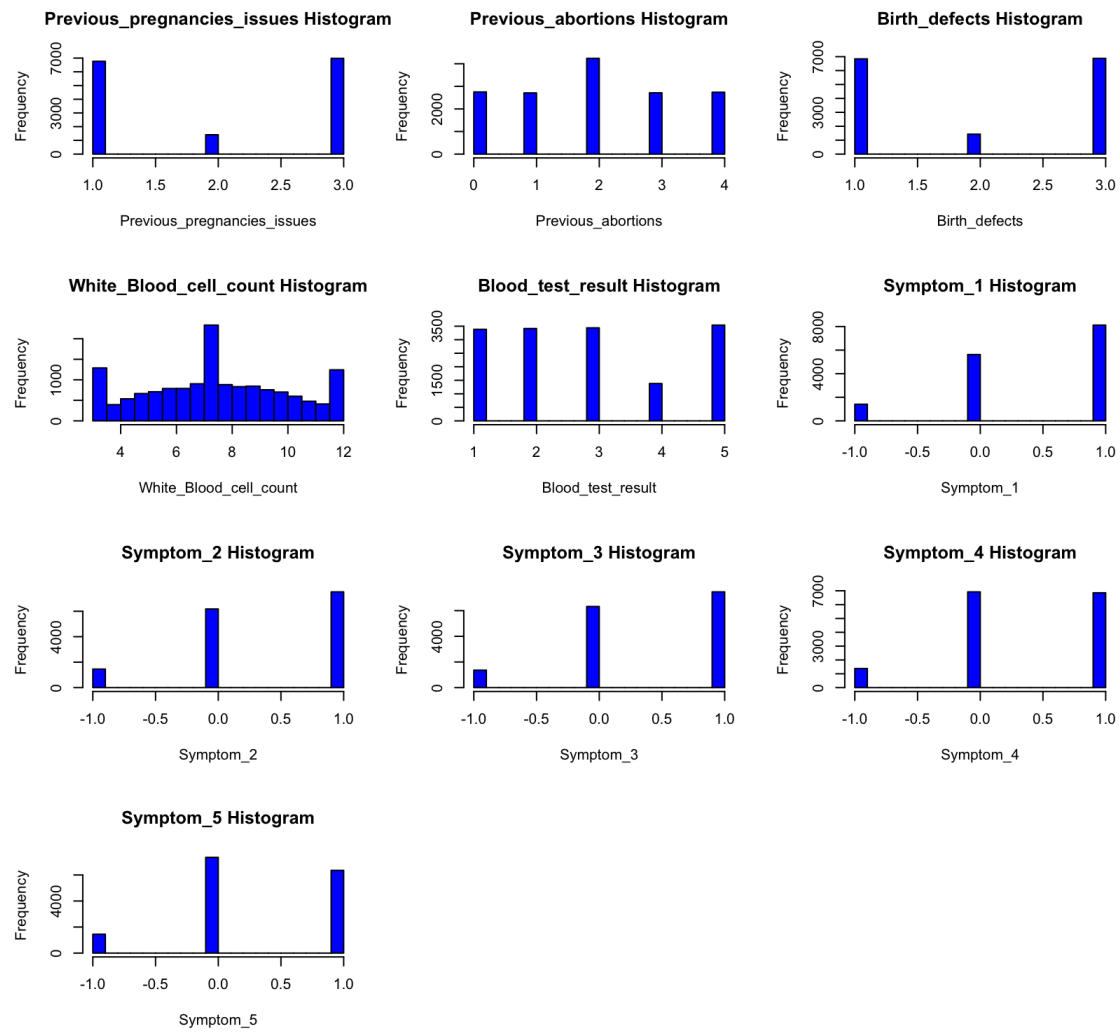
```
Summary Univariate Analysis (15,158 observations)
```

	Type	NA	Blank	Z	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_mothers_side	factor				2								
Genes_fathers_side	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
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	factor				3								
Follow_up	factor				3								
Gender	factor				4								
Autopsy_birth_defect	factor				5								
Folic_acid_concep...	factor				3								
HO_maternal_illness	factor				3								
HO_radiation_expo...	factor				5								
HO_substance_abuse	factor				5								
Assisted_conception	factor				3								
Previous_pregnanc...	factor				3								
Previous_abortions	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								

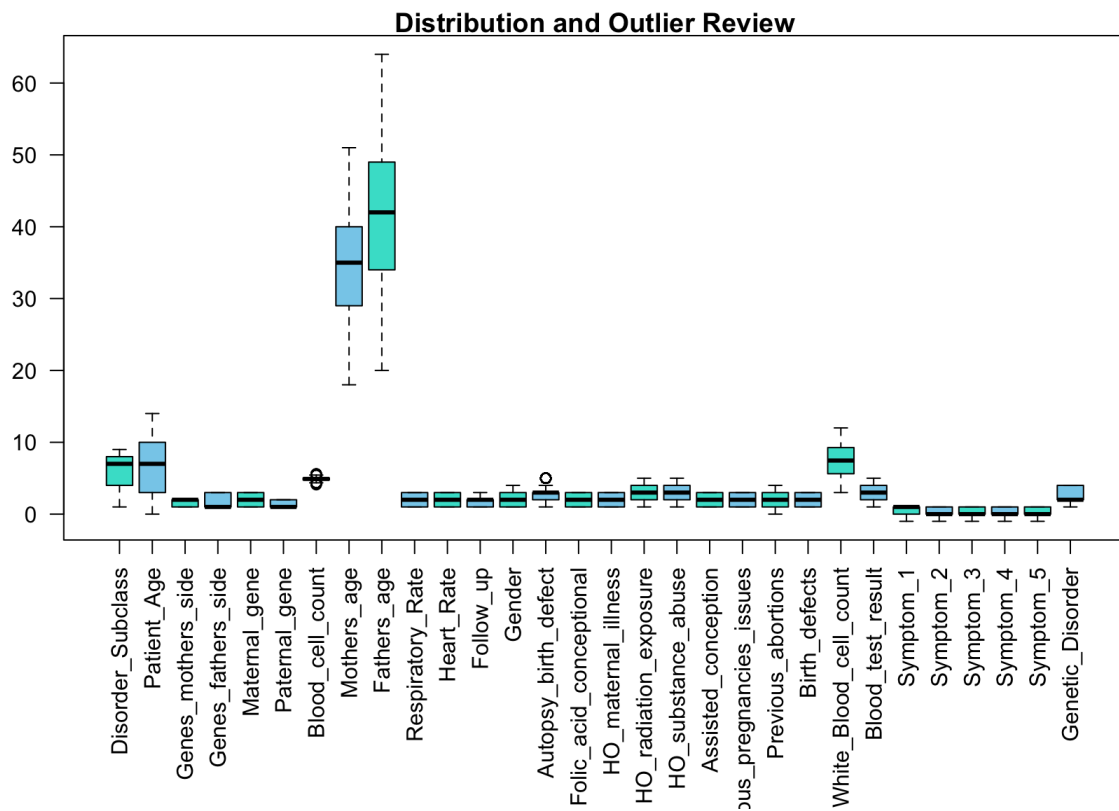
```
#summary(train_df)
```

```
# Generate histograms across predictors and target
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"
  )
```





```
# Generate boxplot(s)
par(mar = c(10, 2, 1, 1))
boxplot(train_df, las = 2, col = c("turquoise", "skyblue"), main = "Distribution and Outlier Review", ylab = "Frequency")
```

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)

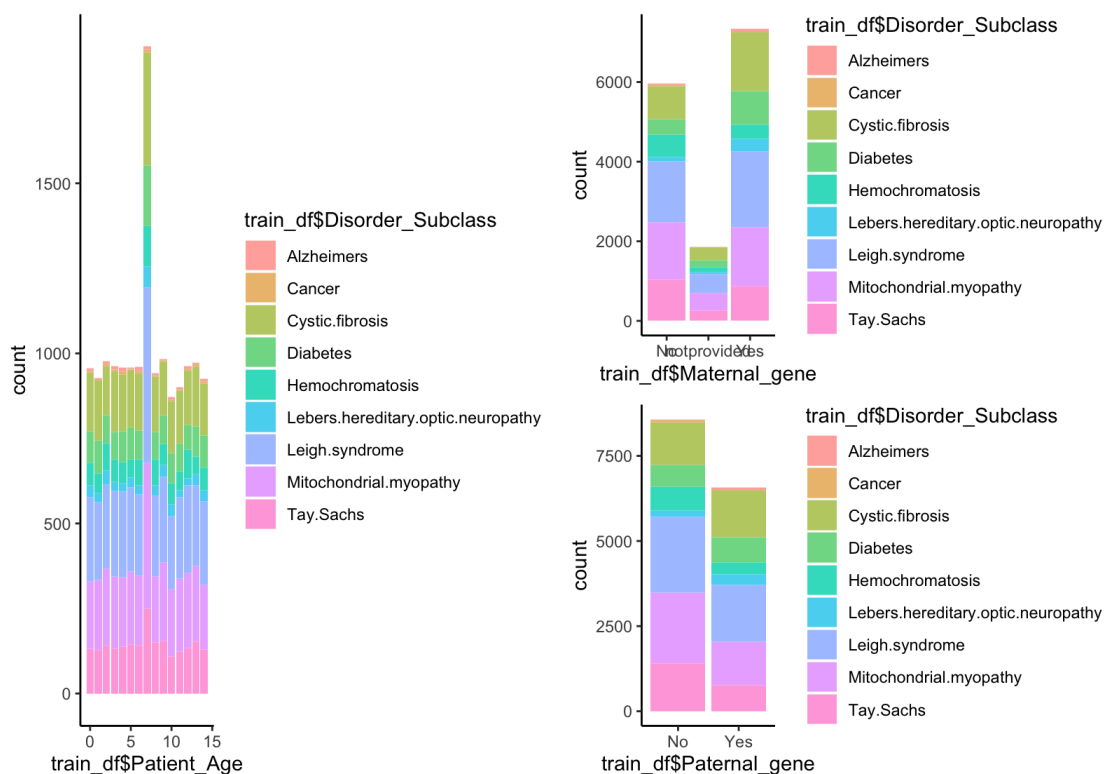
# Shorten variable names for ease of reviewing output matrix
rn <- rownames(cscorr)
for (n in 1:length(rownames(cscorr))) {
  rn[n] <- paste(rownames(cscorr)[n], " (", AscToChar(64 + n), ")", sep = "")
  rownames(cscorr)[n] <- paste(AscToChar(64 + n))
}
for (n in 1:length(colnames(cscorr))) {
  colnames(cscorr)[n]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

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

```
Disorder_Subclass (A)
Genes_mothers_side (B)
Genes_fathers_side (C)
Maternal_gene (D)
Paternal_gene (E)
Respiratory_Rate (F)
Heart_Rate (G)
Follow_up (H)
Gender (I)
Autopsy_birth_defect (J)
Folic_acid_conceptional (K)
HO_maternal_illness (L)
HO_radiation_exposure (M)
HO_substance_abuse (N)
Assisted_conception (O)
Previous_pregnancies_issues (P)
Birth_defects (Q)
Blood_test_result (R)
Genetic_Disorder (S)
```

	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	0.02	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	0.01	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	0.02	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	0.04	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	0.02	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	0.02	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	0.02	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	0.04	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	0.02	0.032	0.051	0.045	0.04	0.035	0.028	0.036	0.04	0.04
J	0.02	0.010	0.016	0.035	0.025	0.019	0.023	0.04	0.02	1.00	0.030	0.022	0.028	0.03	0.021	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	0.03	1.000	0.020	0.049	0.04	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	0.02	0.020	1.000	0.048	0.04	0.043	0.042	0.032	0.04	0.05
M	0.02	0.016	0.030	0.052	0.008	0.030	0.047	0.03	0.04	0.03	0.049	0.048	1.000	0.03	0.046	0.047	0.052	0.04	0.04
N	0.02	0.011	0.015	0.043	0.016	0.035	0.032	0.04	0.04	0.03	0.042	0.037	0.026	1.00	0.033	0.050	0.049	0.03	0.03
O	0.02	0.003	0.013	0.055	0.003	0.035	0.055	0.04	0.03	0.02	0.028	0.043	0.046	0.03	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	0.02	0.032	0.042	0.047	0.05	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	0.03	0.030	0.032	0.052	0.05	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	0.03	0.042	0.041	0.037	0.03	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	0.03	0.035	0.046	0.042	0.03	0.034	0.030	0.053	0.04	1.00

```
# Per hypothesis, relate (visualize) target with maternal and paternal genes to understand more direct relationship
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
```



Predictor Transformations (e.g., PCA)

Modeling

```
# 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$Disorder_Subclass, train_df[ , !sapply(train_df, is.factor)], data.frame(predict(dummies, newdata = train_df)))
dummies <- dummyVars(Disorder_Subclass ~. , data = test_df[ , sapply(test_df, is.factor)])
test_df <- cbind(Disorder_Subclass = test_df$Disorder_Subclass, test_df[ , !sapply(test_df, is.factor)], data.frame(predict(dummies, newdata = test_df)))

# Create common control for models
set.seed(1)
fit_control <- trainControl(method = "cv",
                           savePredictions = "all",
                           classProbs = TRUE,
                           summaryFunction = multiClassSummary)
```

Linear Discriminate Analysis Model

```
# Train LDA model
set.seed(476)
lda_fit <- train(x = train_df[ , -target_col, drop = FALSE],
               y = train_df$Disorder_Subclass,
               method = "lda",
               preProc = c("center", "scale"),
               metric = "ROC",
               trControl = fit_control)
lda_cm <- confusionMatrix(lda_fit, norm = "none")
lda_cm
```

Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

		Reference					
Prediction		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers		0	0	0	0	0	
0	0						
Cancer		0	61	0	30	0	
0	0						
Cystic.fibrosis		0	0	2028	0	50	
0	0						
Diabetes		103	8	0	1240	0	
0	0						
Hemochromatosis		0	0	26	0	85	
0	0						
Lebers.hereditary.optic.neuropathy		9	0	32	29	0	
242	26						
Leigh.syndrome		4	0	141	79	14	
219	2867						
Mitochondrial.myopathy		3	5	57	17	88	
25	1022						
Tay.Sachs		0	0	338	0	796	
0	0						

		Reference	
Prediction		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		0	351
Diabetes		0	0
Hemochromatosis		0	88
Lebers.hereditary.optic.neuropathy		1	0
Leigh.syndrome		1890	60
Mitochondrial.myopathy		1471	143
Tay.Sachs		0	1510

Accuracy (average) : 0.627

```
# Check variable importance
lda_imp <- varImp(lda_fit, scale = FALSE)
lda_imp
```

ROC curve variable importance

variables are sorted by maximum importance across the classes
only 20 most important variables shown (out of 73)

	Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis
Genetic_Disorder.Multifactorial.genetic.inheritance.disorders	0.933	0.533	0.933	0.933	0.933
Genetic_Disorder.Single.gene.inheritance.diseases	0.956	0.500	0.951	0.500	0.500
Genetic_Disorder.Mitochondrial.genetic.inheritance.disorders	0.500	0.500	0.500	0.952	0.953
Symptom_5	0.931	0.931	0.931	0.931	0.931
Symptom_3	0.914	0.914	0.914	0.914	0.914
Symptom_4	0.884	0.884	0.884	0.884	0.884
Symptom_2	0.844	0.844	0.844	0.844	0.844
Symptom_1	0.808	0.808	0.808	0.808	0.808
Genes_mothers_side.Yes	0.787	0.787	0.787	0.787	0.787
Genes_mothers_side.No	0.787	0.787	0.787	0.787	0.787
Paternal_gene.Yes	0.774	0.774	0.774	0.774	0.774
Paternal_gene.No	0.774	0.774	0.774	0.774	0.774
Genes_fathers_side.Yes	0.764	0.764	0.764	0.764	0.764
Genes_fathers_side.No	0.757	0.757	0.757	0.757	0.757
Maternal_gene.Yes	0.737	0.737	0.737	0.737	0.737
Maternal_gene.No	0.711	0.711	0.711	0.711	0.711
Fathers_age	0.570	0.570	0.570	0.570	0.570
Patient_Age	0.562	0.562	0.562	0.562	0.562
Follow_up.High	0.556	0.556	0.556	0.556	0.556
Blood_test_result.inconclusive	0.558	0.558	0.558	0.558	0.558
	Lebers.hereditary.optic.neuropathy		Leigh.syndrome		Mitochondria
1.myopathy Tay.Sachs					
Genetic_Disorder.Multifactorial.genetic.inheritance.disorders			0.933		0.933
0.966 0.933					
Genetic_Disorder.Single.gene.inheritance.diseases			0.500		0.953
0.956 0.956					
Genetic_Disorder.Mitochondrial.genetic.inheritance.disorders			0.950		0.500
0.500 0.500					
Symptom_5			0.931		0.931
0.931 0.632					
Symptom_3			0.914		0.914
0.914 0.629					
Symptom_4			0.884		0.884
0.884 0.606					
Symptom_2			0.844		0.844
0.844 0.593					
Symptom_1			0.808		0.808
0.808 0.586					
Genes_mothers_side.Yes			0.787		0.787
0.787 0.578					
Genes_mothers_side.No			0.787		0.787
0.787 0.578					
Paternal_gene.Yes			0.774		0.774
0.774 0.587					
Paternal_gene.No			0.774		0.774
0.774 0.587					
Genes_fathers_side.Yes			0.764		0.764
0.764 0.595					
Genes_fathers_side.No			0.757		0.757
0.757 0.600					
Maternal_gene.Yes			0.737		0.737
0.737 0.550					
Maternal_gene.No			0.711		0.711
0.711 0.544					
Fathers_age			0.570		0.570
0.570 0.530					
Patient_Age			0.562		0.562
0.562 0.524					
Follow_up.High			0.556		0.556
0.561 0.513					
Blood_test_result.inconclusive			0.558		0.558
0.558 0.526					

Logistic Regression Model

```
# Train LR model
set.seed(476)
invisible(capture.output(
  lr_fit <- train(x = train_df[, -target_col, drop = FALSE],
    y = train_df$Disorder_Subclass,
    method = "multinom",
    metric = "ROC",
    trControl = fit_control)
))
lr_cm <- confusionMatrix(lr_fit, norm = "none")
lr_cm
```

Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

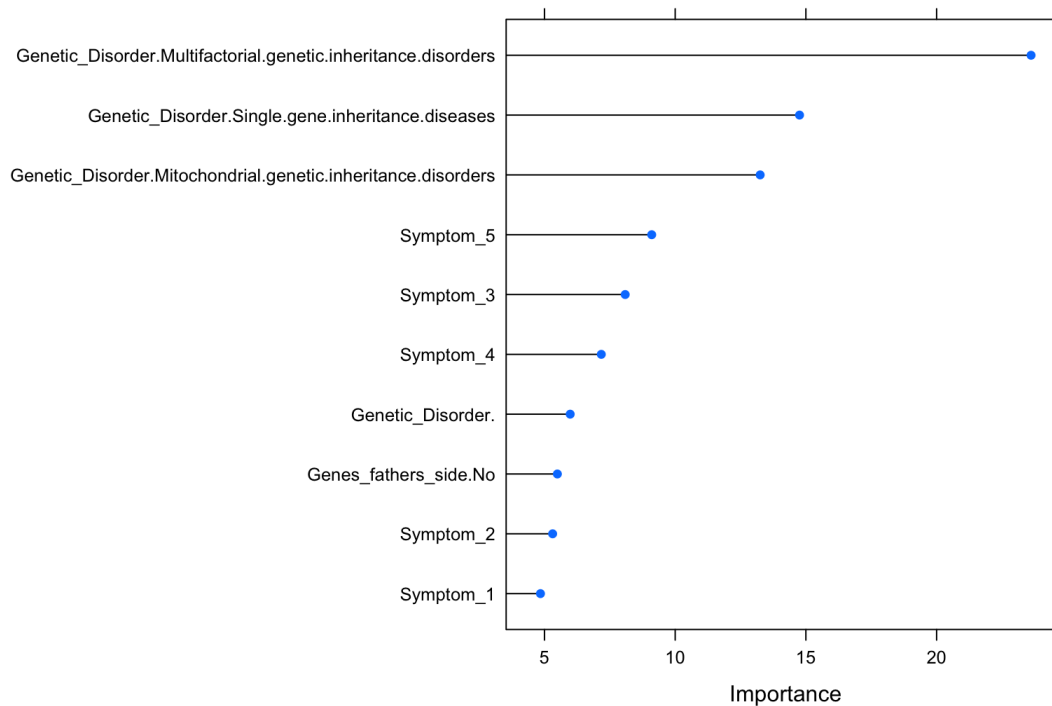
Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers		0	0	1	2	0	
0	0						
Cancer		0	36	0	9	0	
0	0						
Cystic.fibrosis		10	0	2120	67	54	
29	71						
Diabetes		104	33	10	1268	0	
0	4						
Hemochromatosis		0	0	39	0	120	
0	3						
Lebers.hereditary.optic.neuropathy		1	0	0	1	0	
246	29						
Leigh.syndrome		3	0	60	35	24	
172	2595						
Mitochondrial.myopathy		1	4	45	7	57	
36	1190						
Tay.Sachs		0	1	347	6	778	
3	23						

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		23	407
Diabetes		6	5
Hemochromatosis		5	160
Lebers.hereditary.optic.neuropathy		0	0
Leigh.syndrome		1772	66
Mitochondrial.myopathy		1517	86
Tay.Sachs		39	1428

Accuracy (average) : 0.6155

```
# Check variable importance
lr_imp <- varImp(lr_fit, scale = FALSE)
plot(lr_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



Nearest shrunken Centroids Model

```
# Train NSC model
set.seed(476)
invisible(capture.output(
  nsc_fit <- train(x = train_df[, -target_col, drop = FALSE],
    y = train_df$Disorder_Subclass,
    method = "pam",
    preProc = c("center", "scale"),
    tuneGrid = data.frame(threshold = seq(0, 25, length = 30)),
    metric = "ROC",
    trControl = fit_control)
))
nsc_cm <- confusionMatrix(nsc_fit, norm = "none")
nsc_cm
```

Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

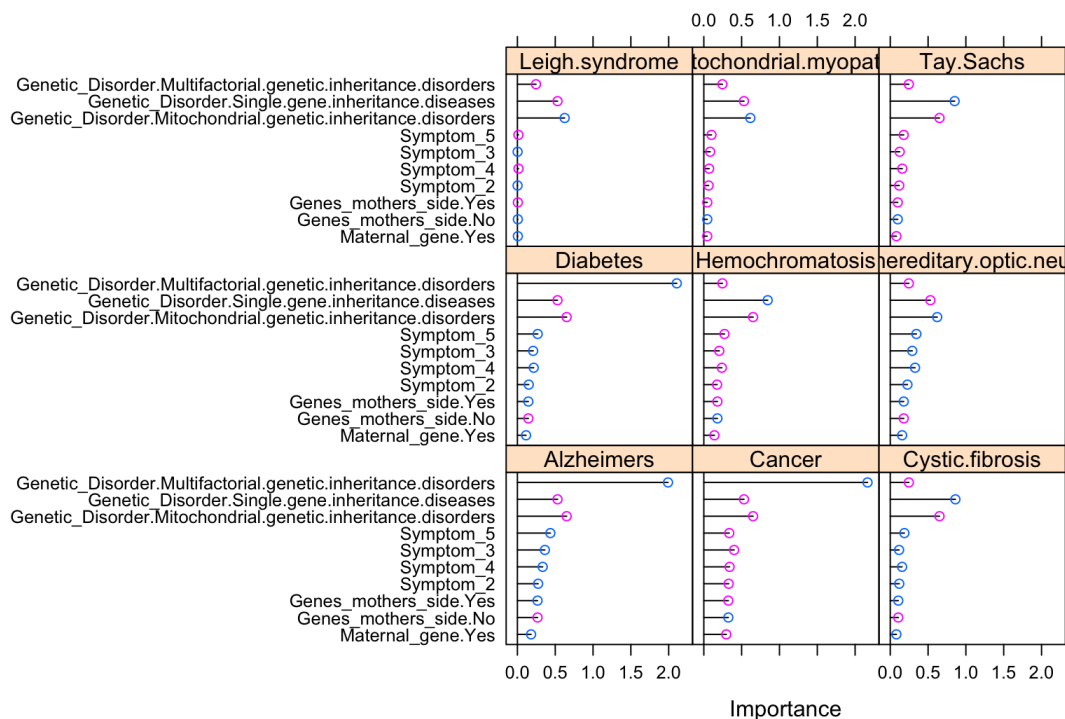
Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers							
0	0		0		0		0
Cancer							
0	0		0		0		0
Cystic.fibrosis							
3	0	1	0	2124	0		210
Diabetes							
0	0	103	69	0	1270		0
Hemochromatosis							
0	0	0	0	0	0		0
Lebers.hereditary.optic.neuropathy							
0	0	0	0	0	0		0
Leigh.syndrome							
478	3578	14	0	207	120		52
Mitochondrial.myopathy							
5	337	1	5	22	5		50
Tay.Sachs							
0	0	0	0	269	0		721

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		0	756
Diabetes		0	0
Hemochromatosis		0	0
Lebers.hereditary.optic.neuropathy		0	0
Leigh.syndrome		2849	129
Mitochondrial.myopathy		513	74
Tay.Sachs		0	1193

Accuracy (average) : 0.5725

```
# Check variable importance
nsc_imp <- varImp(nsc_fit, scale = FALSE)
plot(nsc_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



Random Forest Model

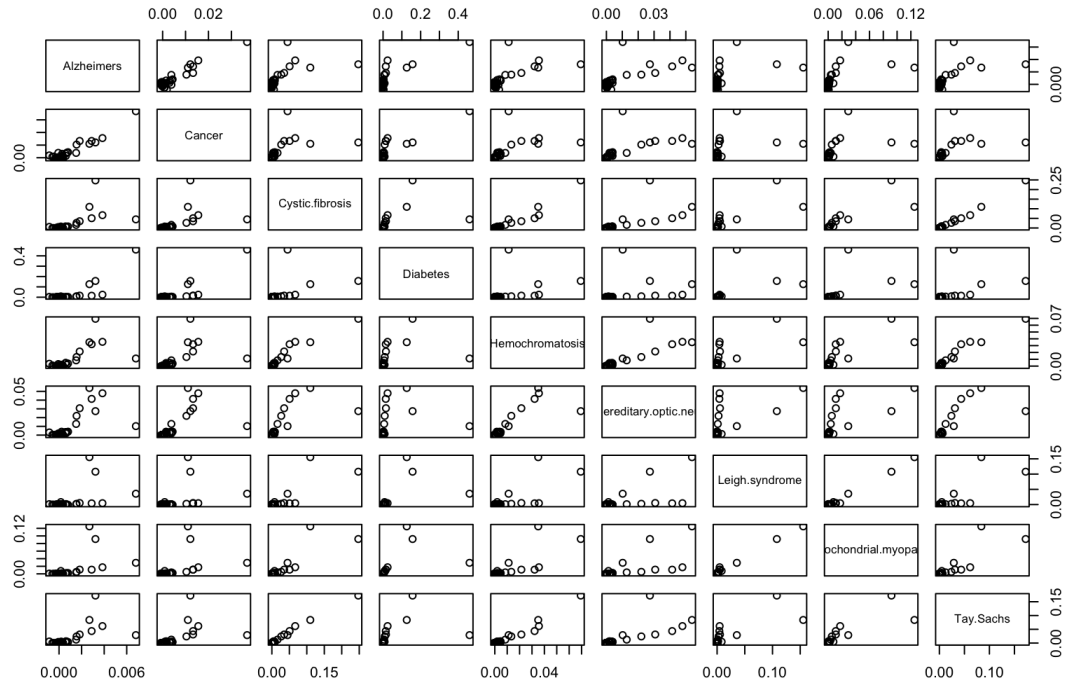
```
# Create Random Forest weight vector based on class priors
priors <- as.list(prop.table(table(train_df$Disorder_Subclass)))
wts <- data.frame(Disorder_Subclass = train_df$Disorder_Subclass, w = 0.0)
for (n in 1:length(priors))
  wts[wts$Disorder_Subclass == names(priors[n]), ]$w <- priors[[n]]

# Train the model (using defaults)
rf_fit <- randomForest(x = train_df[, -target_col, drop = FALSE],
  y = train_df$Disorder_Subclass,
  xtest = test_df[, -target_col, drop = FALSE],
  ytest = test_df$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$confusion)[n], " (", AscToChar(64 + n), ")", sep = "")
for (n in 1:length(rownames(rf_fit$test$confusion)))
  colnames(rf_fit$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))

# Check variable importance
rf_imp <- varImp(rf_fit, scale = FALSE)
plot(rf_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



CART Model

```
# Train CART model
set.seed(476)
cart_fit <- train(x = train_df[, -target_col, drop = FALSE],
  y = train_df$Disorder_Subclass,
  method = "rpart",
  tuneLength = 30,
  metric = "ROC",
  trControl = fit_control)
cart_cm <- confusionMatrix(cart_fit, norm = "none")
cart_cm
```


Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

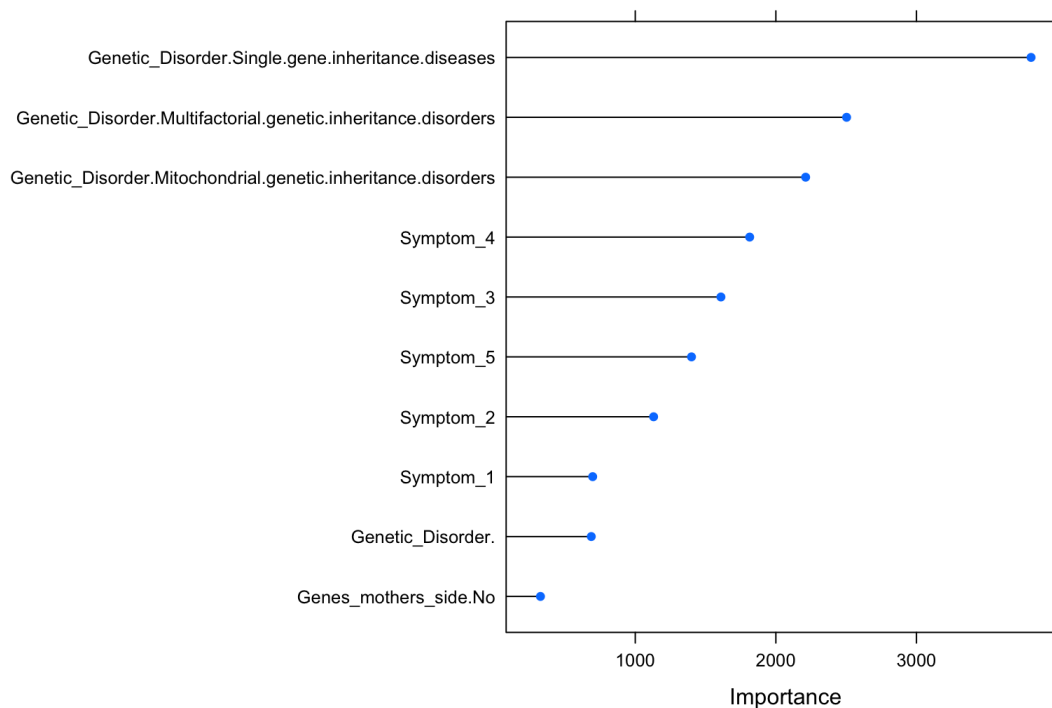
Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers	0	0	0	0	0	0	
Cancer	0	0	62	0	13	0	
Cystic.fibrosis	59	6	0	2208	56	35	
Diabetes	6	108	7	13	1267	0	
Hemochromatosis	0	0	0	4	0	168	
Lebers.hereditary.optic.neuropathy	37	0	0	0	0	0	
Leigh.syndrome	2529	2	0	91	35	15	
Mitochondrial.myopathy	1272	3	4	50	24	60	
Tay.Sachs	12	0	1	256	0	755	

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		25	249
Diabetes		1	0
Hemochromatosis		0	124
Lebers.hereditary.optic.neuropathy		1	0
Leigh.syndrome		1646	60
Mitochondrial.myopathy		1658	123
Tay.Sachs		31	1596

Accuracy (average) : 0.6413

```
# Check variable importance
cart_imp <- varImp(cart_fit, scale = FALSE)
plot(cart_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



Bagged Trees Model

```
# Train BT model
bt_fit <- train(x = train_df[, -target_col, drop = FALSE],
  y = train_df$Disorder_Subclass,
  method = "treebag",
  metric = "ROC",
  importance = TRUE,
  trControl=fit_control)
bt_cm <- confusionMatrix(bt_fit, norm = "none")
bt_cm
```

Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

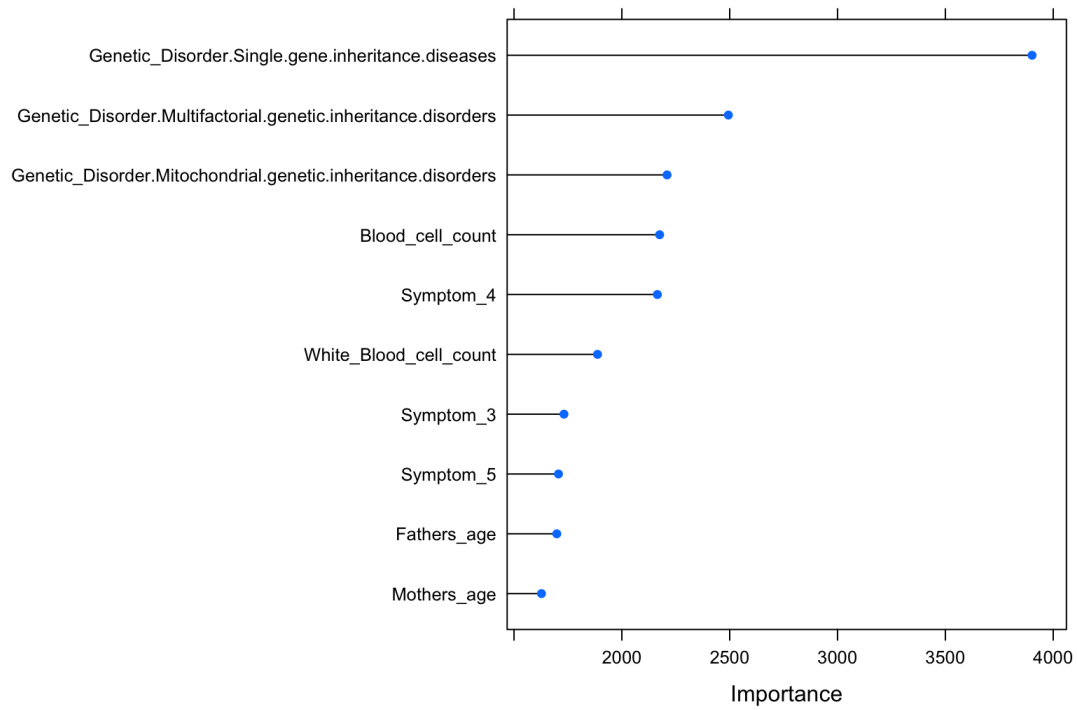
Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers							
0	0	9	0	0	8	0	
Cancer		0	59	0	5	0	
0	0						
Cystic.fibrosis		7	0	2248	43	15	
16	65						
Diabetes		101	10	29	1278	0	
11	8						
Hemochromatosis		0	2	21	0	286	
0	6						
Lebers.hereditary.optic.neuropathy		0	0	3	4	0	
278	45						
Leigh.syndrome		2	0	81	39	19	
163	2375						
Mitochondrial.myopathy		0	0	48	16	37	
17	1386						
Tay.Sachs		0	3	192	2	676	
1	30						

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		27	162
Diabetes		2	0
Hemochromatosis		19	330
Lebers.hereditary.optic.neuropathy		5	0
Leigh.syndrome		1489	52
Mitochondrial.myopathy		1771	72
Tay.Sachs		49	1536

Accuracy (average) : 0.6492

```
# Check variable importance
bt_imp <- varImp(bt_fit, scale = FALSE)
plot(bt_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



KNN Model

```
# Train KNN model
set.seed(476)
knn_fit <- train(x = train_df[, -target_col, drop = FALSE],
  y = train_df$Disorder_Subclass,
  method = "knn",
  metric = "ROC",
  trControl = fit_control)
knn_cm <- confusionMatrix(knn_fit, norm = "none")
knn_cm
```

Cross-Validated (10 fold) Confusion Matrix

(entries are un-normalized aggregated counts)

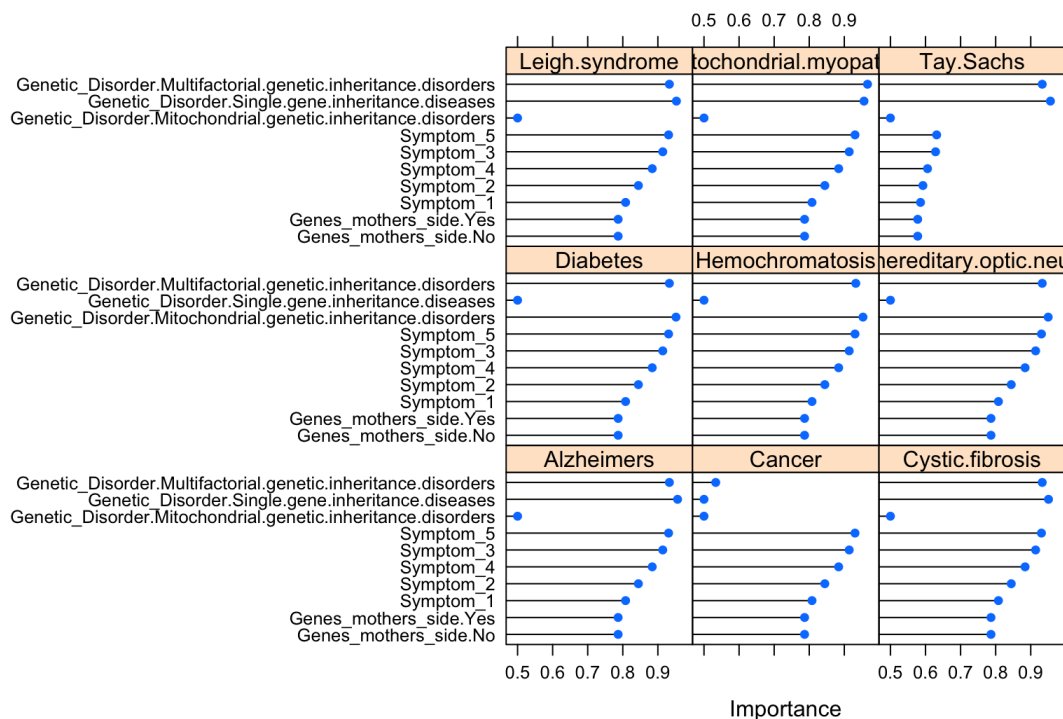
Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers	0	0	0	1	1	0	
Cancer	0	0	0	0	0	0	
Cystic.fibrosis	328	21	10	826	262	185	
Diabetes	92	26	0	118	156	17	
Hemochromatosis	50	2	6	95	23	82	
Lebers.hereditary.optic.neuropathy	22	2	0	18	12	2	
Leigh.syndrome	1754	35	20	657	510	252	
Mitochondrial.myopathy	1420	26	24	489	308	231	
Tay.Sachs	249	7	14	418	123	264	

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		1	0
Cystic.fibrosis		223	442
Diabetes		51	34
Hemochromatosis		59	123
Lebers.hereditary.optic.neuropathy		16	2
Leigh.syndrome		1460	512
Mitochondrial.myopathy		1293	517
Tay.Sachs		259	522

Accuracy (average) : 0.306

```
# Check variable importance
knn_imp <- varImp(knn_fit, scale = FALSE)
plot(knn_imp, top = 10, main = "Top 10 Variables")
```

Top 10 Variables



Model Validation / Evaluation

```
# Validate models
if (lda_eval) {
  print("Linear Disrciminate Analysis")
  lda_pred <- predict(lda_fit, test_df[, -target_col, drop = FALSE])
  lda_pred_cm = confusionMatrix(lda_pred, test_df$Disorder_Subclass)
  lda_pred_cm
}
```

```
[1] "Linear Disrciminate Analysis"
```

Confusion Matrix and Statistics

Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
0	Alzheimers	1	0	0	0	0	
0	Cancer	0	16	0	6	0	
0	Cystic.fibrosis	0	0	505	0	17	
0	Diabetes	26	2	0	310	0	
0	Hemochromatosis	0	0	6	0	28	
0	Lebers.hereditary.optic.neuropathy	1	0	7	8	0	
49	Leigh.syndrome	1	0	42	18	1	
65	Mitochondrial.myopathy	0	0	13	6	16	
7	Tay.Sachs	0	0	82	0	196	
0							

Prediction	Reference	
	Mitochondrial.myopathy	Tay.Sachs
Alzheimers	0	0
Cancer	0	0
Cystic.fibrosis	0	91
Diabetes	0	0
Hemochromatosis	0	24
Lebers.hereditary.optic.neuropathy	1	0
Leigh.syndrome	467	18
Mitochondrial.myopathy	372	35
Tay.Sachs	0	370

Overall Statistics

Accuracy : 0.625
 95% CI : (0.61, 0.641)
 No Information Rate : 0.258
 P-Value [Acc > NIR] : <0.0000000000000002

Kappa : 0.535

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity	0.034483	0.88889	0.771	0.8908	0.1085
Specificity	1.000000	0.99841	0.965	0.9919	0.9915
Pos Pred Value	1.000000	0.72727	0.824	0.9172	0.4828
Neg Pred Value	0.992600	0.99947	0.953	0.9890	0.9383
Prevalence	0.007662	0.00476	0.173	0.0919	0.0682
Detection Rate	0.000264	0.00423	0.133	0.0819	0.0074
Detection Prevalence	0.000264	0.00581	0.162	0.0893	0.0153
Balanced Accuracy	0.517241	0.94365	0.868	0.9413	0.5500
	Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs	
Sensitivity		0.4050	0.732	0.4429	
Specificity		0.9934	0.782	0.8873	
Pos Pred Value		0.6712	0.539	0.5284	
Neg Pred Value		0.9806	0.893	0.8481	
Prevalence		0.0320	0.258	0.2219	
Detection Rate		0.0129	0.189	0.0983	
Detection Prevalence		0.0193	0.351	0.1860	
Balanced Accuracy		0.6992	0.757	0.6651	

```
if (lr_eval) {
  print("Logistic Regression")
  lr_pred <- predict(lr_fit, test_df[, -target_col, drop = FALSE])
  lr_pred_cm = confusionMatrix(lr_pred, test_df$Disorder_Subclass)
  lr_pred_cm
}
```

Confusion Matrix and Statistics

		Reference					
Prediction		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers		0	0	0	0	0	
0	0						
Cancer		0	12	0	4	0	
0	0						
Cystic.fibrosis		1	0	541	22	16	
7	23						
Diabetes		27	6	0	313	0	
0	0						
Hemochromatosis		0	0	8	0	30	
0	0						
Lebers.hereditary.optic.neuropathy		0	0	3	2	0	
54	7						
Leigh.syndrome		1	0	13	2	1	
47	572						
Mitochondrial.myopathy		0	0	11	5	8	
13	367						
Tay.Sachs		0	0	79	0	203	
0	9						

		Reference	
Prediction		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		13	120
Diabetes		0	0
Hemochromatosis		4	36
Lebers.hereditary.optic.neuropathy		0	0
Leigh.syndrome		340	13
Mitochondrial.myopathy		469	27
Tay.Sachs		14	342

Overall Statistics

Accuracy : 0.616
 95% CI : (0.601, 0.632)
 No Information Rate : 0.258
 P-Value [Acc > NIR] : <0.0000000000000002

Kappa : 0.527

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity	0.00000	0.66667	0.826	0.8994	0.11628
Specificity	1.00000	0.99894	0.935	0.9904	0.98639
Pos Pred Value	NaN	0.75000	0.728	0.9046	0.38462
Neg Pred Value	0.99234	0.99841	0.963	0.9898	0.93849
Prevalence	0.00766	0.00476	0.173	0.0919	0.06816
Detection Rate	0.00000	0.00317	0.143	0.0827	0.00793
Detection Prevalence	0.00000	0.00423	0.196	0.0914	0.02061
Balanced Accuracy	0.50000	0.83280	0.881	0.9449	0.55133
	Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs	
Sensitivity		0.4463	0.585	0.558	
Specificity		0.9967	0.851	0.854	
Pos Pred Value		0.8182	0.578	0.521	
Neg Pred Value		0.9820	0.855	0.871	
Prevalence		0.0320	0.258	0.222	
Detection Rate		0.0143	0.151	0.124	
Detection Prevalence		0.0174	0.261	0.238	
Balanced Accuracy		0.7215	0.718	0.706	

```
if (nsc_eval) {  
  print("Nearest Shrunken Centroids")  
  nsc_pred <- predict(nsc_fit, test_df[, -target_col, drop = FALSE])  
  nsc_pred_cm = confusionMatrix(nsc_pred, test_df$Disorder_Subclass)  
  nsc_pred_cm  
}
```

```
[1] "Nearest Shrunken Centroids"
```


Confusion Matrix and Statistics		Reference					
Prediction		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers	0	0	0	0	0	0	
Cancer	0	0	0	0	0	0	
Cystic.fibrosis	0	0	0	528	0	43	
Diabetes	0	27	18	0	316	0	
Hemochromatosis	0	0	0	0	0	0	
Lebers.hereditary.optic.neuropathy	0	0	0	0	0	0	
Leigh.syndrome	120	2	0	57	32	7	
Mitochondrial.myopathy	901	0	0	5	0	10	
Tay.Sachs	77	0	0	65	0	198	
	0						

Prediction		Reference	
		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		0	176
Diabetes		0	0
Hemochromatosis		0	0
Lebers.hereditary.optic.neuropathy		0	0
Leigh.syndrome		730	37
Mitochondrial.myopathy		110	16
Tay.Sachs		0	309

Overall Statistics

Accuracy : 0.572
95% CI : (0.556, 0.588)
No Information Rate : 0.258
P-Value [Acc > NIR] : <0.0000000000000002

Kappa : 0.461

Mcnemar's Test P-Value : NA

Statistics by Class:		Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity		0.00000	0.00000	0.806	0.9080	0.0000
Specificity		1.00000	1.00000	0.930	0.9869	1.0000
Pos Pred Value		NaN	NaN	0.707	0.8753	NaN
Neg Pred Value		0.99234	0.99524	0.958	0.9907	0.9318
Prevalence		0.00766	0.00476	0.173	0.0919	0.0682
Detection Rate		0.00000	0.00000	0.139	0.0835	0.0000
Detection Prevalence		0.00000	0.00000	0.197	0.0954	0.0000
Balanced Accuracy		0.50000	0.50000	0.868	0.9475	0.5000
		Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs	
Sensitivity			0.000	0.921	0.1310	
Specificity			1.000	0.649	0.9630	
Pos Pred Value			NaN	0.478	0.5023	
Neg Pred Value			0.968	0.959	0.7953	
Prevalence			0.032	0.258	0.2219	
Detection Rate			0.000	0.238	0.0291	
Detection Prevalence			0.000	0.498	0.0579	
Balanced Accuracy			0.500	0.785	0.5470	

```

if (rf_eval) {
  print("Random Forest")
  rf_fit
}

```

[1] "Random Forest"

```
Call:
randomForest(x = train_df[, -target_col, drop = FALSE], y = train_df$Disorder_Subclass, xtest = test_df[, -target_col,
drop = FALSE], ytest = test_df$Disorder_Subclass, weights = 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: 37%

Confusion matrix:

	Class A	Class B	Class C	Class D	Class E	Class F	Class G	Class H	Class I	class.error
Alzheimers (A)	0	0	10	103	0	0	6	0	0	1.00
Cancer (B)	0	0	0	69	0	0	1	4	0	1.00
Cystic.fibrosis (C)	0	0	2304	0	0	0	172	16	130	0.12
Diabetes (D)	0	0	50	1270	0	0	68	7	0	0.09
Hemochromatosis (E)	0	0	16	0	0	0	28	67	922	1.00
Lebers.hereditary.optic.neuropathy (F)	0	0	26	1	0	0	455	4	0	1.00
Leigh.syndrome (G)	0	0	16	1	0	0	3014	884	0	0.23
Mitochondrial.myopathy (H)	0	0	1	0	0	0	2076	1276	9	0.62
Tay.Sachs (I)	0	0	216	0	0	0	81	121	1734	0.19

Test set error rate: 36%

Confusion matrix:

	Class A	Class B	Class C	Class D	Class E	Class F	Class G	Class H	Class I	class.error
Alzheimers (A)	0	0	1	27	0	0	1	0	0	1.00
Cancer (B)	0	0	0	18	0	0	0	0	0	1.00
Cystic.fibrosis (C)	0	0	565	0	0	0	48	3	39	0.14
Diabetes (D)	0	0	14	316	0	0	17	1	0	0.09
Hemochromatosis (E)	0	0	4	0	0	0	4	13	237	1.00
Lebers.hereditary.optic.neuropathy (F)	0	0	4	0	0	0	116	1	0	1.00
Leigh.syndrome (G)	0	0	5	0	0	0	787	185	1	0.20
Mitochondrial.myopathy (H)	0	0	0	0	0	0	537	302	1	0.64
Tay.Sachs (I)	0	0	50	0	0	0	23	28	437	0.19

```
if (cart_eval) {
  print("CART")
  # Validate model vs. test data
  cart_pred <- predict(cart_fit, test_df[, -target_col, drop = FALSE])
  cart_pred_cm = confusionMatrix(cart_pred, test_df$Disorder_Subclass)
  cart_pred_cm
}
```

```
[1] "CART"
```

Confusion Matrix and Statistics

Prediction		Reference					
		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
0	Alzheimers	0	0	0	0	0	
0	Cancer	0	15	0	4	0	
0	Cystic.fibrosis	0	0	549	18	5	
2	Diabetes	28	3	13	316	0	
4	Hemochromatosis	0	0	0	0	42	
0	Lebers.hereditary.optic.neuropathy	0	0	0	0	0	
49	Leigh.syndrome	1	0	21	9	2	
64	Mitochondrial.myopathy	0	0	12	1	15	
2	Tay.Sachs	0	0	60	0	194	
0							

Prediction	Reference	
	Mitochondrial.myopathy	Tay.Sachs
Alzheimers	0	0
Cancer	0	0
Cystic.fibrosis	8	62
Diabetes	1	0
Hemochromatosis	0	36
Lebers.hereditary.optic.neuropathy	4	0
Leigh.syndrome	410	13
Mitochondrial.myopathy	417	40
Tay.Sachs	0	387

Overall Statistics

Accuracy : 0.651
 95% CI : (0.636, 0.666)
 No Information Rate : 0.258
 P-Value [Acc > NIR] : <0.0000000000000002

Kappa : 0.568

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity	0.00000	0.83333	0.838	0.9080	0.1628
Specificity	1.00000	0.99894	0.963	0.9843	0.9898
Pos Pred Value	NaN	0.78947	0.826	0.8541	0.5385
Neg Pred Value	0.99234	0.99920	0.966	0.9906	0.9417
Prevalence	0.00766	0.00476	0.173	0.0919	0.0682
Detection Rate	0.00000	0.00396	0.145	0.0835	0.0111
Detection Prevalence	0.00000	0.00502	0.176	0.0978	0.0206
Balanced Accuracy	0.50000	0.91614	0.901	0.9462	0.5763
	Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs	
Sensitivity		0.4050	0.704	0.496	
0.719					
Specificity		0.9967	0.815	0.890	
0.922					
Pos Pred Value		0.8033	0.570	0.562	
0.604					
Neg Pred Value		0.9807	0.888	0.861	
0.952					
Prevalence		0.0320	0.258	0.222	
0.142					
Detection Rate		0.0129	0.182	0.110	
0.102					
Detection Prevalence		0.0161	0.319	0.196	
0.169					
Balanced Accuracy		0.7008	0.760	0.693	
0.821					

```

if (bt_eval) {
  print("Bagged Trees")
  bt_pred <- predict(bt_fit, test_df[, -target_col, drop = FALSE])
  bt_pred_cm = confusionMatrix(bt_pred, test_df$Disorder_Subclass)
  bt_pred_cm
}

```

Confusion Matrix and Statistics

		Reference					
Prediction		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
Alzheimers		7	0	0	5	0	
0	0						
Cancer		0	16	0	0	0	
0	0						
Cystic.fibrosis		1	0	552	16	5	
4	18						
Diabetes		21	2	7	318	0	
2	3						
Hemochromatosis		0	0	4	0	63	
0	0						
Lebers.hereditary.optic.neuropathy		0	0	0	2	0	
66	9						
Leigh.syndrome		0	0	26	4	2	
43	615						
Mitochondrial.myopathy		0	0	11	2	8	
6	327						
Tay.Sachs		0	0	55	1	180	
0	6						

		Reference	
Prediction		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		7	43
Diabetes		2	0
Hemochromatosis		7	68
Lebers.hereditary.optic.neuropathy		1	0
Leigh.syndrome		389	15
Mitochondrial.myopathy		420	17
Tay.Sachs		14	395

Overall Statistics

Accuracy : 0.648
 95% CI : (0.632, 0.663)
 No Information Rate : 0.258
 P-Value [Acc > NIR] : <0.0000000000000002

Kappa : 0.567

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity	0.24138	0.88889	0.843	0.9138	0.2442
Specificity	0.99867	1.00000	0.970	0.9892	0.9776
Pos Pred Value	0.58333	1.00000	0.854	0.8958	0.4437
Neg Pred Value	0.99417	0.99947	0.967	0.9913	0.9465
Prevalence	0.00766	0.00476	0.173	0.0919	0.0682
Detection Rate	0.00185	0.00423	0.146	0.0840	0.0166
Detection Prevalence	0.00317	0.00423	0.171	0.0938	0.0375
Balanced Accuracy	0.62002	0.94444	0.906	0.9515	0.6109

	Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs
Sensitivity	0.5455	0.629	0.500	
Specificity	0.9967	0.829	0.874	
Pos Pred Value	0.8462	0.562	0.531	
Neg Pred Value	0.9852	0.865	0.860	
Prevalence	0.0320	0.258	0.222	
Detection Rate	0.0174	0.162	0.111	
Detection Prevalence	0.0206	0.289	0.209	
Balanced Accuracy	0.7711	0.729	0.687	

```
if (knn_eval) {  
  print("KNN")  
  knn_pred <- predict(knn_fit, test_df[, -target_col, drop = FALSE])  
  knn_pred_cm = confusionMatrix(knn_pred, test_df$Disorder_Subclass)  
  knn_pred_cm  
}
```

```
[1] "KNN"
```

Confusion Matrix and Statistics

		Reference					
Prediction		Alzheimers	Cancer	Cystic.fibrosis	Diabetes	Hemochromatosis	Lebers.hereditary.optic.neur
opathy Leigh.syndrome							
0	Alzheimers	0	0	0	0	0	
0	Cancer	0	0	0	0	0	
0	Cystic.fibrosis	7	2	198	61	44	
11	Diabetes	5	0	31	49	2	
6	Hemochromatosis	0	3	21	5	25	
1	Lebers.hereditary.optic.neuropathy	0	0	2	1	2	
1	Leigh.syndrome	11	7	168	111	66	
66	Mitochondrial.myopathy	5	5	118	86	63	
34	Tay.Sachs	1	1	117	35	56	
2							

		Reference	
Prediction		Mitochondrial.myopathy	Tay.Sachs
Alzheimers		0	0
Cancer		0	0
Cystic.fibrosis		65	105
Diabetes		15	6
Hemochromatosis		19	36
Lebers.hereditary.optic.neuropathy		3	0
Leigh.syndrome		368	129
Mitochondrial.myopathy		310	104
Tay.Sachs		60	158

Overall Statistics

Accuracy : 0.316
95% CI : (0.301, 0.331)
No Information Rate : 0.258
P-Value [Acc > NIR] : 0.00000000000000101

Kappa : 0.137

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Alzheimers	Class: Cancer	Class: Cystic.fibrosis	Class: Diabetes	Class: Hemochromatosis
Sensitivity	0.00000	0.00000	0.3023	0.1408	0.09690
Specificity	1.00000	1.00000	0.8812	0.9770	0.97250
Pos Pred Value	NaN	NaN	0.3474	0.3828	0.20492
Neg Pred Value	0.99234	0.99524	0.8579	0.9182	0.93639
Prevalence	0.00766	0.00476	0.1731	0.0919	0.06816
Detection Rate	0.00000	0.00000	0.0523	0.0129	0.00661
Detection Prevalence	0.00000	0.00000	0.1506	0.0338	0.03223
Balanced Accuracy	0.50000	0.50000	0.5917	0.5589	0.53470
	Class: Lebers.hereditary.optic.neuropathy	Class: Leigh.syndrome	Class: Mitochondrial.myopathy	Class: Tay.Sachs	
Sensitivity		0.008264	0.466	0.3690	
0.2937					
Specificity		0.996725	0.670	0.7382	
0.8981					
Pos Pred Value		0.076923	0.330	0.2868	
0.3231					
Neg Pred Value		0.968187	0.783	0.8040	
0.8847					
Prevalence		0.031968	0.258	0.2219	
0.1421					
Detection Rate		0.000264	0.120	0.0819	
0.0417					
Detection Prevalence		0.003435	0.365	0.2856	
0.1292					
Balanced Accuracy		0.502495	0.568	0.5536	
0.5959					

```

## Plot the ROC curve for the hold-out set
if (lda_eval) {
  lda_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(lda_pred))
  plot.roc(lda_roc$rocs[[1]], type = "s", col = 'red', legacy.axes = TRUE,
    main = "Compare ROC Curves for All Models")
}

if (lr_eval) {
  lr_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(lr_pred))
  plot.roc(lr_roc$rocs[[1]], type = "s", add = TRUE, col = 'green', legacy.axes = TRUE)
}

if (nsc_eval) {
  nsc_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(nsc_pred))
  plot.roc(nsc_roc$rocs[[1]], type = "s", add = TRUE, col = 'blue', legacy.axes = TRUE)
}

if (rf_eval) {
  rf_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(rf_fit$test$predicted))
  plot.roc(rf_roc$rocs[[1]], type = "s", col = 'orange', add = TRUE, legacy.axes = TRUE)
  #par(pty = "s")
  #lines <- sapply(2:length(rocs), function(x) lines.roc(rocs[[x]], col = x))
  #dev <- dev.off()
}

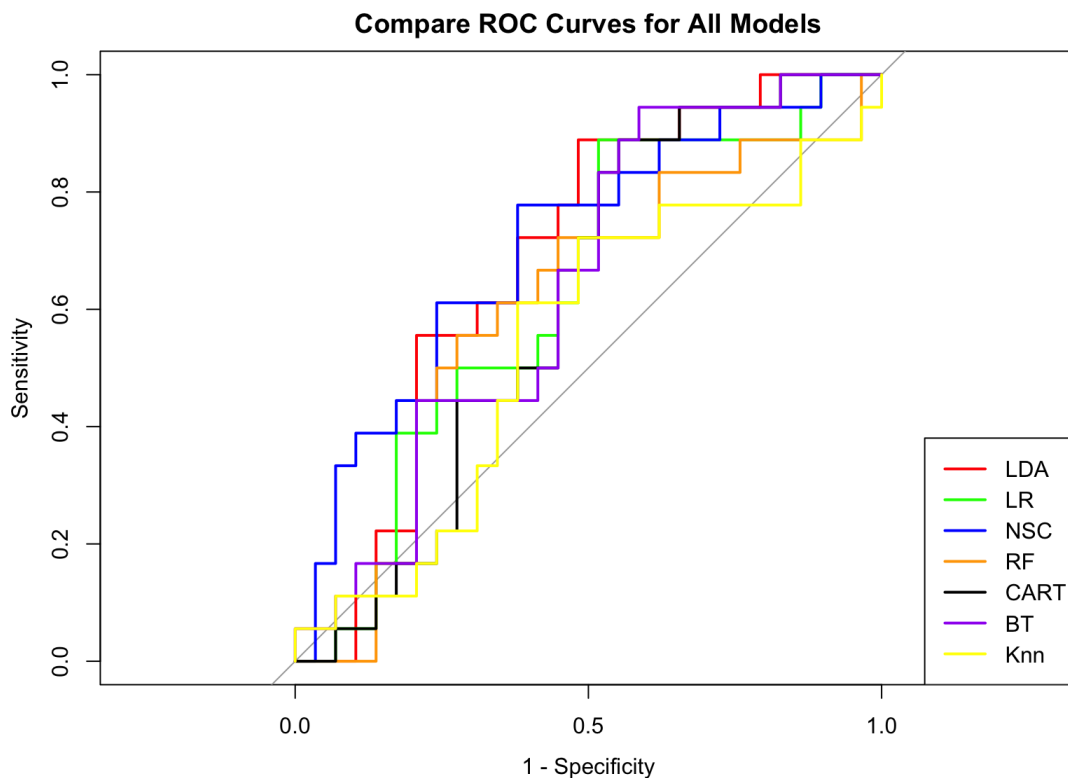
if (cart_eval) {
  cart_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(cart_pred))
  plot.roc(cart_roc$rocs[[1]], type = "s", col = 'black', add = TRUE, legacy.axes = TRUE)
}

if (bt_eval) {
  bt_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(bt_pred))
  plot.roc(bt_roc$rocs[[1]], type = "s", col = 'purple', add = TRUE, legacy.axes = TRUE)
}

if (knn_eval) {
  knn_roc <- multiclass.roc(response = test_df$Disorder_Subclass,
    predictor = order(knn_pred))
  plot.roc(knn_roc$rocs[[1]], type = "s", col = 'yellow', add = TRUE, legacy.axes = TRUE)
}

if (lda_eval | lr_eval | nsc_eval | rf_eval | cart_eval | bt_eval | knn_eval)
  legend("bottomright", legend = c("LDA", "LR", "NSC", "RF", "CART", "BT", "Knn"),
    col = c("red", "green", "blue", "orange", "black", "purple", "yellow"), lwd = 2)

```



-
1. University of San Diego, eoosandiego@ucsd.edu (mailto:eoosandiego@ucsd.edu)↵
 2. University of San Diego, sbhattarai@ucsd.edu (mailto:sbhattarai@ucsd.edu)↵
 3. University of San Diego, dfriesen@ucsd.edu (mailto:dfriesen@ucsd.edu)↵