ADS502 Group Project

Francisco Hernandez, Jason Morfin, Brendan Robinson, Aaron Gabriel

2025-04-02

```r
threshold <- 0.5
drug_induced_training_data <- drug_induced_training_data[, colMeans(is.na(drug_induced_training_data)) <

get_mode <- function(x) {
    uniq_vals <- unique(x)
    uniq_vals[which.max(tabulate(match(x, uniq_vals)))]
}
categorical_cols <- sapply(drug_induced_training_data, is.character)
drug_induced_training_data[categorical_cols] <- lapply(
    drug_induced_training_data[categorical_cols],
    function(x) {
        x[is.na(x)] <- get_mode(x)
        return(x)
    })
write.csv(drug_induced_training_data, "DIA_trainingset_RDKit_descriptors.csv")
```

Data Cleaning

```r
summary(drug_induced_training_data[, 1:6])
```

Descriptive Statistics

```
##      Label            SMILES             BalabanJ        BertzCT      
##  Min.   :0.0000   Length:477         Min.   :0.986   Min.   :   8.0  
##  1st Qu.:0.0000   Class :character   1st Qu.:1.679   1st Qu.: 493.3  
##  Median :0.0000   Mode  :character   Median :1.964   Median : 712.4  
##  Mean   :0.2474                      Mean   :2.143   Mean   : 738.6  
##  3rd Qu.:0.0000                      3rd Qu.:2.419   3rd Qu.: 943.2  
##  Max.   :1.0000                      Max.   :5.083   Max.   :2430.9  
##       Chi0            Chi0n       
##  Min.   : 3.414   Min.   : 1.725  
##  1st Qu.:13.405   1st Qu.:10.391  
##  Median :17.646   Median :14.184  
##  Mean   :18.130   Mean   :14.372  
##  3rd Qu.:22.052   3rd Qu.:17.730  
##  Max.   :50.120   Max.   :38.475  
```

```
drug_data <- select(drug_induced_training_data, where(is.numeric))
drug_stats <- describe(drug_data)
head(drug_stats, 10)
```

```
##          vars   n   mean     sd median trimmed    mad  min     max   range skew
## Label       1 477   0.25   0.43   0.00    0.19   0.00 0.00    1.00    1.00 1.17
## BalabanJ    2 477   2.14   0.71   1.96    2.05   0.52 0.99    5.08    4.10 1.49
## BertzCT     3 477 738.63 392.97 712.42  720.01 336.93 8.00 2430.93 2422.93 0.79
## Chi0        4 477  18.13   7.25  17.65   17.71   6.41 3.41   50.12   46.71 0.84
## Chi0n       5 477  14.37   6.09  14.18   14.05   5.50 1.73   38.48   36.75 0.81
## Chi0v       6 477  14.89   6.13  14.68   14.58   5.52 1.73   39.84   38.11 0.77
## Chi1        7 477  11.87   4.83  11.77   11.67   4.29 1.73   31.52   29.78 0.66
## Chi1n       8 477   8.43   3.76   8.35    8.26   3.54 0.61   23.17   22.55 0.65
## Chi1v       9 477   8.96   3.81   9.00    8.79   3.39 0.61   24.44   23.83 0.62
## Chi2n      10 477   6.67   3.28   6.36    6.48   3.04 0.25   19.35   19.10 0.75
##          kurtosis    se
## Label       -0.64  0.02
## BalabanJ     2.66  0.03
## BertzCT      1.74 17.99
## Chi0         1.73  0.33
## Chi0n        1.77  0.28
## Chi0v        1.62  0.28
## Chi1         1.38  0.22
## Chi1n        1.18  0.17
## Chi1v        1.13  0.17
## Chi2n        1.05  0.15
```

```
data_quality_report <- data.frame(
  Variable = names(drug_induced_training_data),
  Type = sapply(drug_induced_training_data, class),
  Missing = sapply(drug_induced_training_data, function(x) sum(is.na(x))),
  Complete = sapply(drug_induced_training_data, function(x) sum(!is.na(x))),
  Unique = sapply(drug_induced_training_data, function(x) length(unique(x)))
)
knitr::kable(head(data_quality_report, 40), caption = "Drug Induced Data Quality Report (Preview)")
```

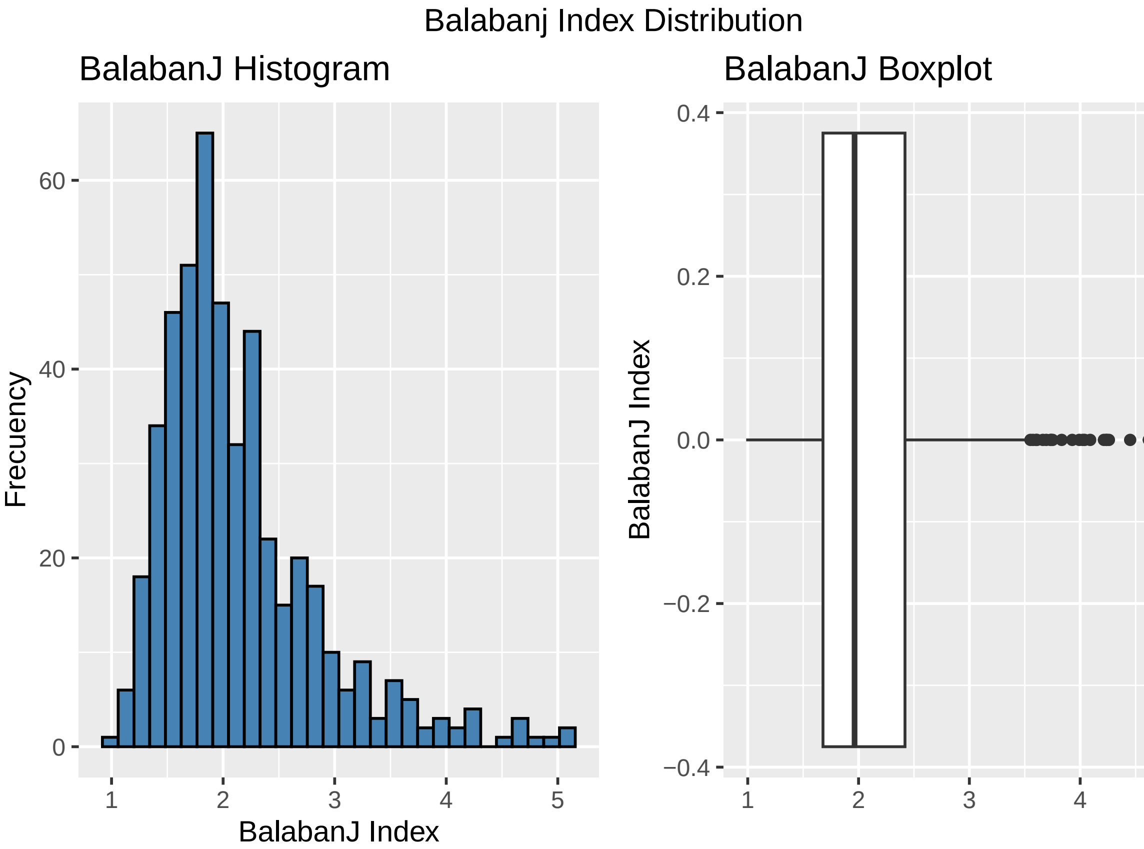
Data Quality Report

Table 1: Drug Induced Data Quality Report (Preview)

	Variable	Type	Missing	Complete	Unique
Label	Label	integer	0	477	2
SMILES	SMILES	character	0	477	477
BalabanJ	BalabanJ	numeric	0	477	406
BertzCT	BertzCT	numeric	0	477	465
Chi0	Chi0	numeric	0	477	382
Chi0n	Chi0n	numeric	0	477	460
Chi0v	Chi0v	numeric	0	477	460
Chi1	Chi1	numeric	0	477	423

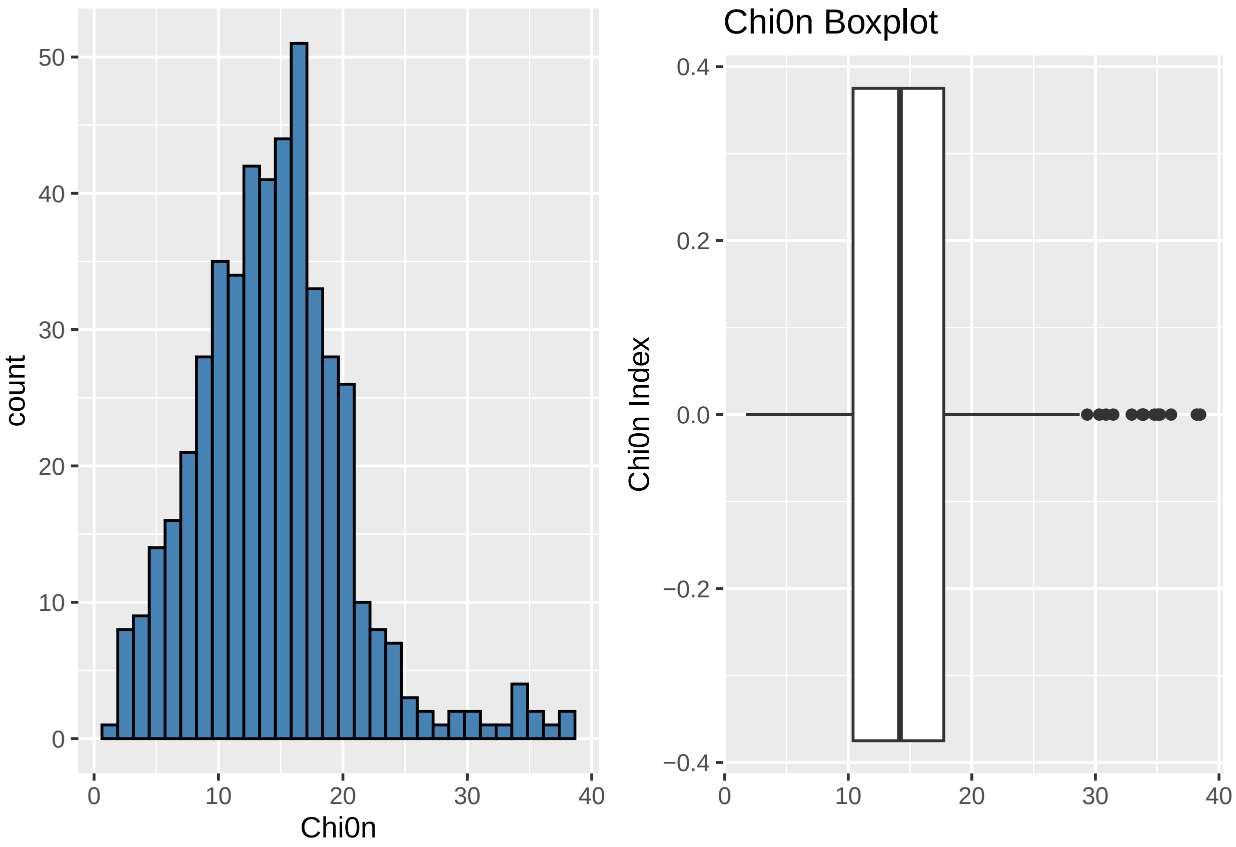
	Variable	Type	Missing	Complete	Unique
Chi1n	Chi1n	numeric	0	477	461
Chi1v	Chi1v	numeric	0	477	466
Chi2n	Chi2n	numeric	0	477	458
Chi2v	Chi2v	numeric	0	477	462
Chi3n	Chi3n	numeric	0	477	456
Chi3v	Chi3v	numeric	0	477	457
Chi4n	Chi4n	numeric	0	477	450
Chi4v	Chi4v	numeric	0	477	447
EState_VSA1	EState_VSA1	numeric	0	477	211
EState_VSA10	EState_VSA10	numeric	0	477	108
EState_VSA11	EState_VSA11	numeric	0	477	5
EState_VSA2	EState_VSA2	numeric	0	477	279
EState_VSA3	EState_VSA3	numeric	0	477	233
EState_VSA4	EState_VSA4	numeric	0	477	244
EState_VSA5	EState_VSA5	numeric	0	477	183
EState_VSA6	EState_VSA6	numeric	0	477	131
EState_VSA7	EState_VSA7	numeric	0	477	122
EState_VSA8	EState_VSA8	numeric	0	477	217
EState_VSA9	EState_VSA9	numeric	0	477	134
ExactMolWt	ExactMolWt	numeric	0	477	460
FractionCSP3	FractionCSP3	numeric	0	477	175
HallKierAlpha	HallKierAlpha	numeric	0	477	247
HeavyAtomCount	HeavyAtomCount	integer	0	477	53
HeavyAtomMolWt	HeavyAtomMolWt	numeric	0	477	420
Ipc	Ipc	numeric	0	477	453
Kappa1	Kappa1	numeric	0	477	450
Kappa2	Kappa2	numeric	0	477	459
Kappa3	Kappa3	numeric	0	477	448
LabuteASA	LabuteASA	numeric	0	477	466
MaxAbsEStateIndex	MaxAbsEStateIndex	numeric	0	477	452
MaxAbsPartialCharge	MaxAbsPartialCharge	numeric	0	477	166
MaxEStateIndex	MaxEStateIndex	numeric	0	477	452

```r
hist <- ggplot(drug_induced_training_data, aes(x=BalabanJ)) + geom_histogram(bins = 30, fill = "steelblu
  labs(title = "BalabanJ Histogram", x = "BalabanJ Index", y = "Frecuency")
boxplot <- ggplot(drug_induced_training_data, aes(x=BalabanJ)) + geom_boxplot() +
labs(title = "BalabanJ Boxplot", x = "", y = "BalabanJ Index")

grid.arrange(hist, boxplot, ncol = 2, top="Balabanj Index Distribution")
```

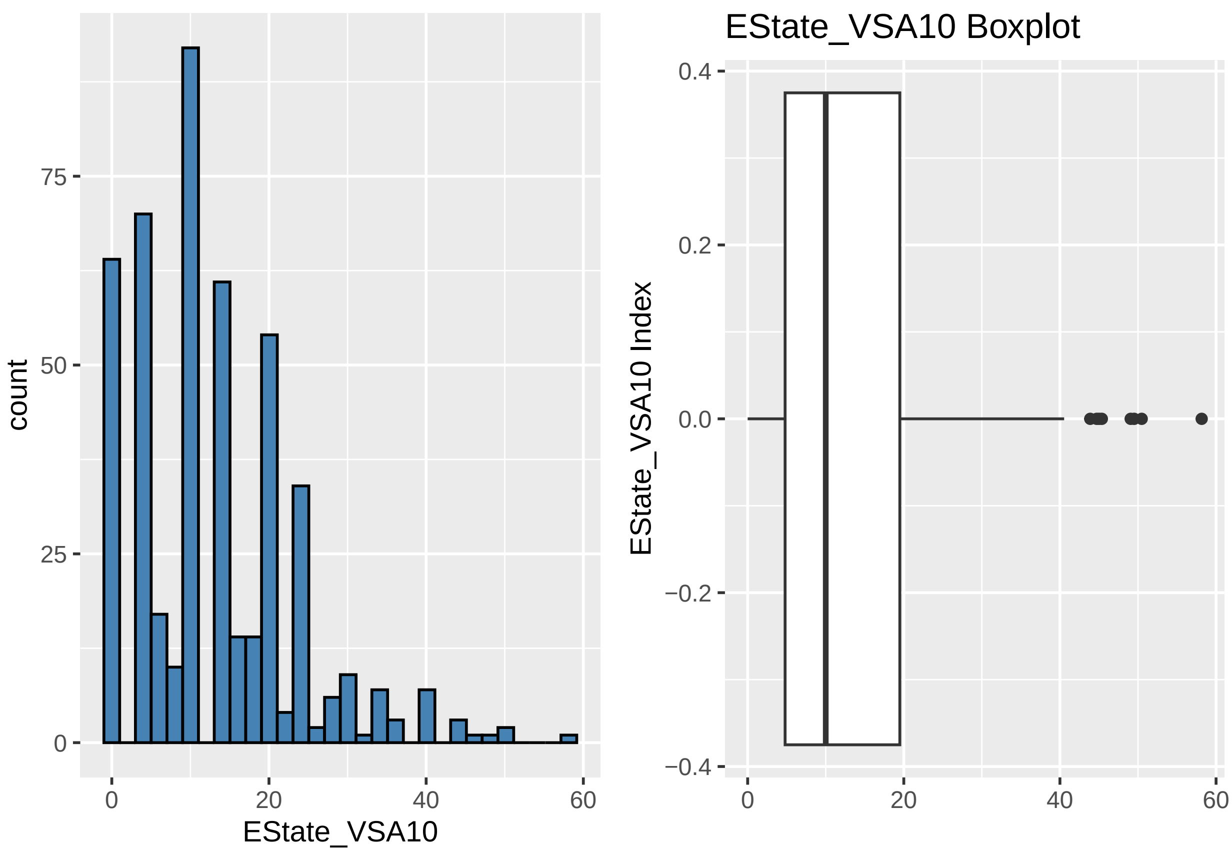


Univariate Analysis

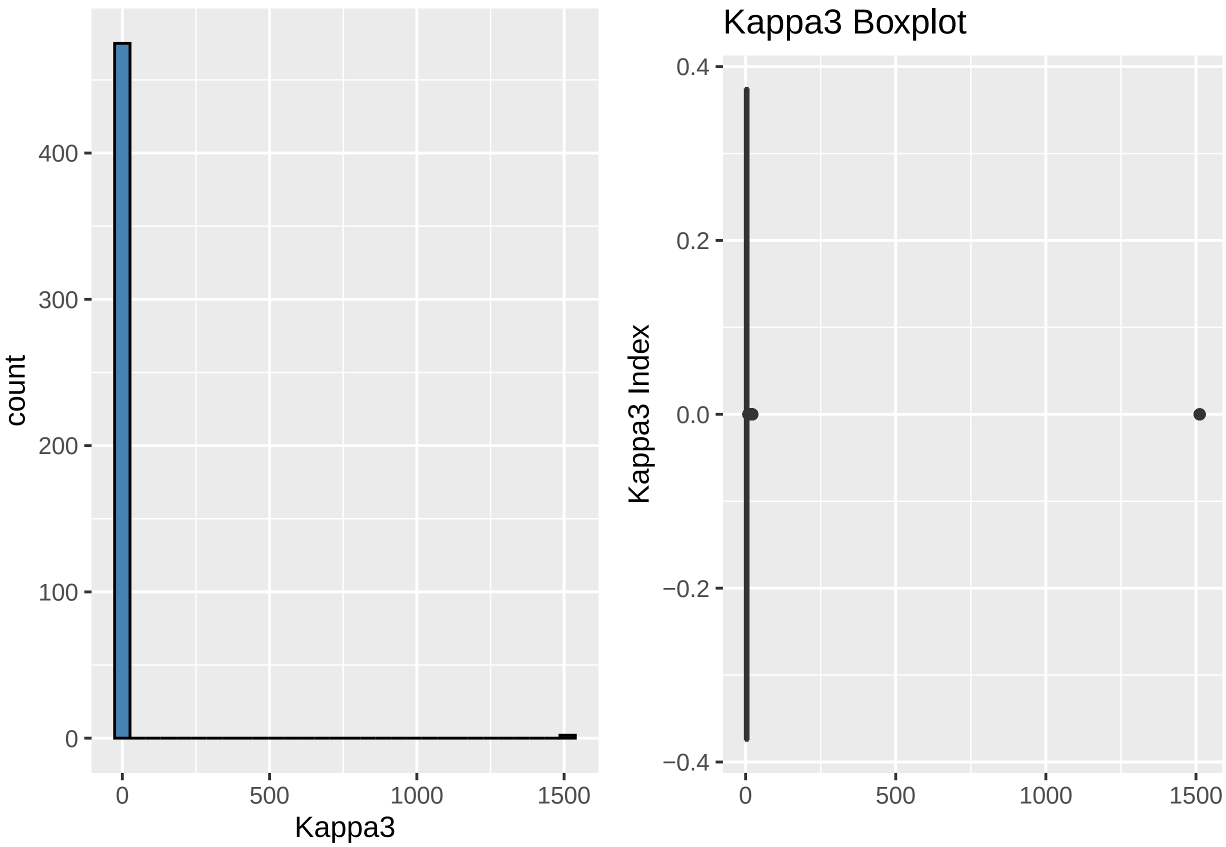
```
hist_2 <- ggplot(drug_induced_training_data, aes(x=ChiOn)) + geom_histogram(bins = 30, fill = "steelblu
boxplot_2 <- ggplot(drug_induced_training_data, aes(x=ChiOn)) + geom_boxplot() +
  labs(title = "ChiOn Boxplot", x = "", y = "ChiOn Index")

grid.arrange(hist_2, boxplot_2, ncol = 2)
```



```
hist_3 <- ggplot(drug_induced_training_data, aes(x=EState_VSA10)) + geom_histogram(bins = 30, fill = "s
boxplot_3 <- ggplot(drug_induced_training_data, aes(x=EState_VSA10)) + geom_boxplot() +
  labs(title = "EState_VSA10 Boxplot", x = "", y = "EState_VSA10 Index")

grid.arrange(hist_3, boxplot_3, ncol = 2)
```

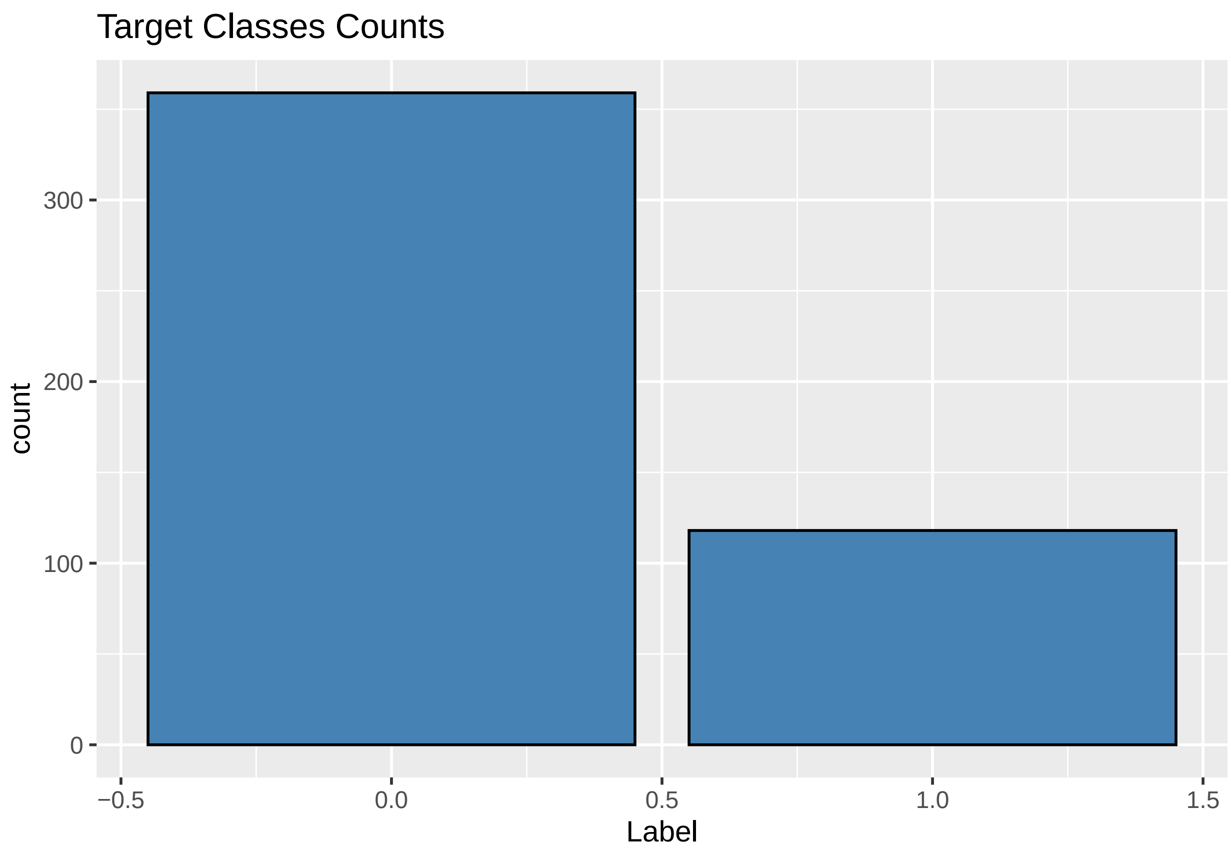


```
hist_4 <- ggplot(drug_induced_training_data, aes(x=Kappa3)) + geom_histogram(bins = 30, fill = "steelblu
boxplot_4 <- ggplot(drug_induced_training_data, aes(x=Kappa3)) + geom_boxplot() +
  labs(title = "Kappa3 Boxplot", x = "", y = "Kappa3 Index")

grid.arrange(hist_4, boxplot_4, ncol = 2)
```



Features are right sweked. Few outliers that do not need to be removed.

```
hist_6 <- ggplot(drug_induced_training_data, aes(x =Label)) +
  geom_bar(fill = "steelblue", color = "black") +
  labs(title = "Target Classes Counts")
hist_6
```



Multivariate Analysis

```
numeric_data <- drug_induced_training_data[, sapply(drug_induced_training_data, is.numeric)]
numeric_data <- numeric_data[, apply(numeric_data, 2, sd, na.rm = TRUE) != 0]
numeric_data <- na.omit(numeric_data)

cor_matrix <- cor(numeric_data)
cor_matrix[lower.tri(cor_matrix, diag = TRUE)] <- NA
cor_long <- melt(cor_matrix, na.rm = TRUE)
top_corr <- cor_long[order(-abs(cor_long$value)), ]

head(top_corr, 10)
```

Top 10 correlated features

```
##                         Var1            Var2     value
## 6877        MaxAbsEStateIndex      MaxEStateIndex 1.0000000
## 24173 NumAromaticCarbocycles          fr_benzene 1.0000000
## 22249                fr_Ar_NH        fr_Nhpyrrole 1.0000000
## 8307              ExactMolWt               MolWt 0.9999970
## 5227                    Chi1      HeavyAtomCount 0.9986496
## 8311            HeavyAtomMolWt               MolWt 0.9984270
## 5427              ExactMolWt      HeavyAtomMolWt 0.9983729
## 11164                   Chi0 NumValenceElectrons 0.9966473
## 5224                    Chi0      HeavyAtomCount 0.9960391
## 6330          HeavyAtomCount           LabuteASA 0.9955411
```

```
print("Pre-balance class cardinality")
```

Class Imbalance

```
## [1] "Pre-balance class cardinality"
```

```
table(drug_induced_training_data$Label)
```

```
##
##   0   1
## 359 118
```

```
# Random Undersampling
balanced_0 <- sample_n(filter(drug_induced_training_data, Label == 0), 118)
filtered_1 <- filter(drug_induced_training_data, Label == 1)
bal_drug_induced_training_data <- rbind(balanced_0, filtered_1)

print("Post-balance class cardinality")
```

```
## [1] "Post-balance class cardinality"
```

```
table(bal_drug_induced_training_data$Label)
```

```
##
##   0   1
## 118 118
```

```
# Drop SMILES feature
bal_drug_induced_training_data <- bal_drug_induced_training_data[, colnames(bal_drug_induced_training_da

for (col_name in colnames(bal_drug_induced_training_data)){
# For binary responses
  if (length(unique(bal_drug_induced_training_data[[col_name]])) == 2){
    bal_drug_induced_training_data[[col_name]] <- as.factor(bal_drug_induced_training_data[[col_name]])
  }

  # For standarization of numeric responses
  if (is.numeric(bal_drug_induced_training_data[[col_name]])){
    if( max(bal_drug_induced_training_data[[col_name]]) != min(bal_drug_induced_training_data[[col_name]
    bal_drug_induced_training_data[[col_name]] <- (bal_drug_induced_training_data[[col_name]] - min(bal_
```

```r
    (max(bal_drug_induced_training_data[[col_name]]) - min(bal_drug_induced_training_data[[col_name]]))
    }
    else{
          bal_drug_induced_training_data[[col_name]] <- 0
    }
  }
}
# Final Dataset
# bal_drug_induced_training_data
```

Pre-Model Data Preparation

```r
# Missing step --> numeric_features need to be obtained from bal_drug_induced_training_data
numeric_cols_indexes <- sapply(drug_induced_training_data, is.numeric)
numeric_features <- drug_induced_training_data[numeric_cols_indexes]

model_all <- lm(Label ~ ., data=numeric_features)  # with all the independent variables in the datafram
# summary(model_all)
coeff <- coefficients(model_all)

# To remove correlated where coefficient = NaN -> Correlation
na_coeff_names <- names(coeff)[is.na(coeff)]
na_coeff_names
```

Feature Selection

```
##  [1] "MaxEStateIndex"      "NumAliphaticRings"   "NumAromaticRings"
##  [4] "NumRadicalElectrons" "NumSaturatedRings"   "RingCount"
##  [7] "SMR_VSA8"            "SlogP_VSA9"          "VSA_EState1"
## [10] "VSA_EState2"         "VSA_EState3"         "VSA_EState4"
## [13] "VSA_EState5"         "VSA_EState6"         "VSA_EState7"
## [16] "fr_COO"              "fr_Nhpyrrole"        "fr_azide"
## [19] "fr_barbitur"         "fr_benzene"          "fr_diazo"
## [22] "fr_isocyan"          "fr_isothiocyan"      "fr_prisulfonamd"
## [25] "fr_thiocyan"
```

Train and Test

```r
# Remove pound if you haven't installed packages below
#loading libraries for modeling and evaluation
# install.packages("hardhat")
# install.packages("parallelly")
# install.packages("caret")

library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
# Drop features with NA coefficients
cleaned_data <- bal_drug_induced_training_data[, !(names(bal_drug_induced_training_data) %in% na_coeff_

# Ensure Label is a factor
cleaned_data$Label <- as.factor(cleaned_data$Label)

# Split data into train (80%) and test (20%)
set.seed(123)
train_index <- createDataPartition(cleaned_data$Label, p = 0.8, list = FALSE)
train_data <- cleaned_data[train_index, ]
test_data <- cleaned_data[-train_index, ]
```

```
# determining constant variables for removal
constant_vars <- c("fr_epoxide", "fr_morpholine", "fr_oxime", "fr_tetrazole")

# removing from the training and test datasets:
train_data_fixed <- train_data[, !(names(train_data) %in% constant_vars)]
test_data_fixed  <- test_data[, !(names(test_data) %in% constant_vars)]

# Remove the constant variable fr_oxazole from both training and test data
train_data_fixed <- train_data_fixed[, !(names(train_data_fixed) %in% c("fr_oxazole"))]
test_data_fixed  <- test_data_fixed[, !(names(test_data_fixed) %in% c("fr_oxazole"))]

# removes factor columns with only one level
train_data_fixed <- train_data_fixed[, sapply(train_data_fixed, function(col) {
  !(is.factor(col) || is.character(col)) || length(unique(col)) > 1
})]
```

```
# Baseline Model - Logistic Regression

# Train logistic regression model
log_model <- glm(Label ~ ., data = train_data_fixed, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Predict probabilities on test data
log_probs <- predict(log_model, newdata = test_data_fixed, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```r
# Convert probabilities to class labels
log_preds <- ifelse(log_probs > 0.5, 1, 0)

# Evaluate model
confusionMatrix(as.factor(log_preds), test_data_fixed$Label, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9 11
##          1 14 12
##
##                Accuracy : 0.4565
##                  95% CI : (0.309, 0.6099)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.7693
##
##                   Kappa : -0.087
##
##  Mcnemar's Test P-Value : 0.6892
##
##             Sensitivity : 0.5217
##             Specificity : 0.3913
##          Pos Pred Value : 0.4615
##          Neg Pred Value : 0.4500
##              Prevalence : 0.5000
##          Detection Rate : 0.2609
##    Detection Prevalence : 0.5652
##       Balanced Accuracy : 0.4565
##
##        'Positive' Class : 1
##
```

```r
# Remove pound if you haven't installed packages below
# install.packages("e1071")
# install.packages("randomForest")
# install.packages("C50")
```

```r
# Train Other Models (Naive Bayes, Random Forest, C5.0)

# Load libraries
library(e1071)
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:psych':
##
##     outlier

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(C50)

# Naive Bayes
nb_model <- naiveBayes(Label ~ ., data = train_data)
nb_preds <- predict(nb_model, newdata = test_data)
confusionMatrix(nb_preds, test_data$Label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 12  4
##          1 11 19
##
##                Accuracy : 0.6739
##                  95% CI : (0.5198, 0.8047)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.01295
##
##                   Kappa : 0.3478
##
##  Mcnemar's Test P-Value : 0.12134
##
##             Sensitivity : 0.5217
##             Specificity : 0.8261
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.6333
##              Prevalence : 0.5000
##          Detection Rate : 0.2609
##    Detection Prevalence : 0.3478
##       Balanced Accuracy : 0.6739
##
##        'Positive' Class : 0
##
```

```r
# Random Forest
rf_model <- randomForest(Label ~ ., data = train_data, ntree = 100)
rf_preds <- predict(rf_model, newdata = test_data)
confusionMatrix(rf_preds, test_data$Label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 17  4
##          1  6 19
##
##                Accuracy : 0.7826
##                  95% CI : (0.6364, 0.8905)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 7.821e-05
##
##                   Kappa : 0.5652
##
##  Mcnemar’s Test P-Value : 0.7518
##
##             Sensitivity : 0.7391
##             Specificity : 0.8261
##          Pos Pred Value : 0.8095
##          Neg Pred Value : 0.7600
##              Prevalence : 0.5000
##          Detection Rate : 0.3696
##    Detection Prevalence : 0.4565
##       Balanced Accuracy : 0.7826
##
##        ’Positive’ Class : 0
##
```

```r
# C5.0
c50_model <- C5.0(Label ~ ., data = train_data)
c50_preds <- predict(c50_model, newdata = test_data)
confusionMatrix(c50_preds, test_data$Label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 13  8
##          1 10 15
##
##                Accuracy : 0.6087
##                  95% CI : (0.4537, 0.7491)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.09196
##
##                   Kappa : 0.2174
##
##  Mcnemar’s Test P-Value : 0.81366
```

```
##
##             Sensitivity : 0.5652
##             Specificity : 0.6522
##          Pos Pred Value : 0.6190
##          Neg Pred Value : 0.6000
##              Prevalence : 0.5000
##          Detection Rate : 0.2826
##    Detection Prevalence : 0.4565
##       Balanced Accuracy : 0.6087
##
##        ’Positive’ Class : 0
##
```

```
# creating a function to extract evaluation metrics from confusion matrix
get_metrics <- function(preds, actual) {
  cm <- confusionMatrix(as.factor(preds), as.factor(actual), positive = "1")
  c(
    Accuracy = cm$overall["Accuracy"],
    Sensitivity = cm$byClass["Sensitivity"],
    Specificity = cm$byClass["Specificity"],
    Precision = cm$byClass["Pos Pred Value"]
  )
}

# building model evaluation table for model comparison
accuracy_summary <- data.frame(
  Model = c("Logistic Regression", "Naive Bayes", "Random Forest", "C5.0"),
  rbind(
    get_metrics(log_preds, test_data$Label),
    get_metrics(nb_preds, test_data$Label),
    get_metrics(rf_preds, test_data$Label),
    get_metrics(c50_preds, test_data$Label)
  )
)

# View result
print(accuracy_summary)
```

```
##                 Model Accuracy.Accuracy Sensitivity.Sensitivity
## 1 Logistic Regression         0.4565217               0.5217391
## 2         Naive Bayes         0.6739130               0.8260870
## 3       Random Forest         0.7826087               0.8260870
## 4                C5.0         0.6086957               0.6521739
##   Specificity.Specificity Precision.Pos.Pred.Value
## 1               0.3913043                0.4615385
## 2               0.5217391                0.6333333
## 3               0.7391304                0.7600000
## 4               0.5652174                0.6000000
```