Machine Learning Model to predict normal and tumour samples

```
# install.packages(c('MLmetrics', 'matrixStats', 'kknn', 'glmnet'))
```

```
suppressPackageStartupMessages(library(tidyverse))
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
suppressMessages(library(kknn))
suppressPackageStartupMessages(library(glmnet))
```

```
# load preprocessed data
counts_data <- read_csv('preprocessed_counts.csv', show_col_types = FALSE)
```

```
## New names:
## * `` -> `...1`
```

```
counts_data <- counts_data |> column_to_rownames(var='...1')

metadata <- read_csv('metadata.csv', show_col_types = F)

dea_results <- read.csv('DEA_results_Tumor_vs_Normal.csv', row.names=1)
```

```
head(counts_data)
```

```
##                 TCGA.VS.A8EI.01A.11R.A37O.07 TCGA.FU.A3HZ.01A.11R.A213.07
## ENSG00000000003                         4189                         4693
## ENSG00000000005                            0                            0
## ENSG00000000419                         2801                         2757
## ENSG00000000457                          447                          696
## ENSG00000000460                          308                          563
## ENSG00000000938                          210                          201
##                 TCGA.DS.A7WH.01A.22R.A352.07 TCGA.EA.A1QS.01A.61R.A22U.07
## ENSG00000000003                         1846                         4503
## ENSG00000000005                            0                            0
## ENSG00000000419                         2735                         3001
## ENSG00000000457                          791                          523
## ENSG00000000460                         1714                          390
## ENSG00000000938                          209                          530
##                 TCGA.EK.A2RM.01A.21R.A18M.07 TCGA.MA.AA3Z.01A.11R.A38B.07
## ENSG00000000003                         4999                         3689
## ENSG00000000005                            0                            0
## ENSG00000000419                         3907                         2895
## ENSG00000000457                          950                          823
## ENSG00000000460                          626                          992
```

```
## ENSG00000000938                         156                         362
##                 TCGA.ZJ.AAX4.01A.11R.A42T.07 TCGA.EA.A5O9.01A.11R.A28H.07
## ENSG00000000003                         6358                       10035
## ENSG00000000005                            0                           0
## ENSG00000000419                         2968                        2151
## ENSG00000000457                          964                         639
## ENSG00000000460                          691                        1168
## ENSG00000000938                          582                         135
##                 TCGA.LP.A7HU.01A.11R.A33Z.07 TCGA.MA.AA3Y.01A.11R.A38B.07
## ENSG00000000003                         5280                        1479
## ENSG00000000005                            0                           0
## ENSG00000000419                         3996                        3158
## ENSG00000000457                          948                         581
## ENSG00000000460                          942                         723
## ENSG00000000938                          252                         429
##                 TCGA.VS.A9UB.01A.22R.A42T.07 TCGA.VS.A954.01A.11R.A38B.07
## ENSG00000000003                         1117                        1822
## ENSG00000000005                            0                           0
## ENSG00000000419                         1640                        3859
## ENSG00000000457                          523                         956
## ENSG00000000460                          628                        1034
## ENSG00000000938                          890                         147
##                 TCGA.Q1.A73S.01A.11R.A33Z.07 TCGA.VS.A9UP.01A.11R.A42T.07
## ENSG00000000003                         6006                        3970
## ENSG00000000005                            0                           0
## ENSG00000000419                         4287                        2201
## ENSG00000000457                          914                        1168
## ENSG00000000460                          822                        1170
## ENSG00000000938                           84                          43
##                 TCGA.EK.A2RC.01A.11R.A18M.07 TCGA.EA.A439.01A.11R.A24H.07
## ENSG00000000003                         2249                        2256
## ENSG00000000005                            0                           0
## ENSG00000000419                         2419                        3821
## ENSG00000000457                          716                         851
## ENSG00000000460                          737                        1138
## ENSG00000000938                          135                         477
##                 TCGA.DS.A7WF.01A.11R.A352.07 TCGA.DS.A1OB.01A.11R.A14Y.07
## ENSG00000000003                         2778                        1147
## ENSG00000000005                            4                           0
## ENSG00000000419                         1929                        1713
## ENSG00000000457                          831                         947
## ENSG00000000460                          707                        1318
## ENSG00000000938                           61                         511
##                 TCGA.PN.A8MA.01A.11R.A36F.07 TCGA.FU.A3TQ.01A.11R.A22U.07
## ENSG00000000003                         3918                        3166
## ENSG00000000005                            0                           0
## ENSG00000000419                         5685                        1562
## ENSG00000000457                          857                        1058
## ENSG00000000460                          940                         880
## ENSG00000000938                          211                         328
##                 TCGA.LP.A4AW.01A.11R.A24H.07 TCGA.WL.A834.01A.11R.A352.07
## ENSG00000000003                         7186                        2629
## ENSG00000000005                            1                           0
## ENSG00000000419                         2757                        4001
```

```
## ENSG00000000457                          535                          1119
## ENSG00000000460                          441                          1190
## ENSG00000000938                          320                           206
##                 TCGA.Q1.A73Q.01A.21R.A32P.07 TCGA.C5.A2LV.01A.11R.A18M.07
## ENSG00000000003                         2125                          3221
## ENSG00000000005                            0                             0
## ENSG00000000419                         2326                          3574
## ENSG00000000457                         1398                           496
## ENSG00000000460                         1374                          1015
## ENSG00000000938                          296                           716
##                 TCGA.C5.A1BF.01B.11R.A13Y.07 TCGA.DG.A2KM.01A.11R.A180.07
## ENSG00000000003                         3298                          1822
## ENSG00000000005                            0                             0
## ENSG00000000419                         2236                          2771
## ENSG00000000457                          689                           702
## ENSG00000000460                          511                           441
## ENSG00000000938                          220                          2185
##                 TCGA.C5.A1MN.01A.11R.A14Y.07 TCGA.JW.A852.01A.11R.A352.07
## ENSG00000000003                         3727                          2280
## ENSG00000000005                            0                             0
## ENSG00000000419                         4885                          2345
## ENSG00000000457                          466                           602
## ENSG00000000460                          685                           368
## ENSG00000000938                          141                            85
##                 TCGA.EK.A3GK.01A.11R.A213.07 TCGA.JW.A5VK.01A.11R.A28H.07
## ENSG00000000003                         3765                          5432
## ENSG00000000005                            0                             0
## ENSG00000000419                         4208                          4782
## ENSG00000000457                          971                           765
## ENSG00000000460                          854                           659
## ENSG00000000938                          317                           123
##                 TCGA.C5.A902.01A.11R.A37O.07 TCGA.EA.A50E.01A.21R.A26T.07
## ENSG00000000003                         2333                          3148
## ENSG00000000005                            0                            80
## ENSG00000000419                         3601                          3058
## ENSG00000000457                          890                           608
## ENSG00000000460                          709                           704
## ENSG00000000938                          207                           478
##                 TCGA.DG.A2KJ.01A.11R.A32Y.07 TCGA.C5.A1BJ.01A.11R.A13Y.07
## ENSG00000000003                         2892                          1799
## ENSG00000000005                            0                             0
## ENSG00000000419                         3557                          2845
## ENSG00000000457                          730                           658
## ENSG00000000460                          106                           769
## ENSG00000000938                           90                           249
##                 TCGA.MA.AA3W.01A.11R.A38B.07 TCGA.C5.A8ZZ.01A.11R.A37O.07
## ENSG00000000003                         1861                          2010
## ENSG00000000005                            0                             0
## ENSG00000000419                         2316                          1937
## ENSG00000000457                          942                           672
## ENSG00000000460                          830                           561
## ENSG00000000938                          557                            89
##                 TCGA.DR.A0ZL.01A.11R.A10U.07 TCGA.MY.A5BF.11A.11R.A26T.07
## ENSG00000000003                         3495                          4606
```

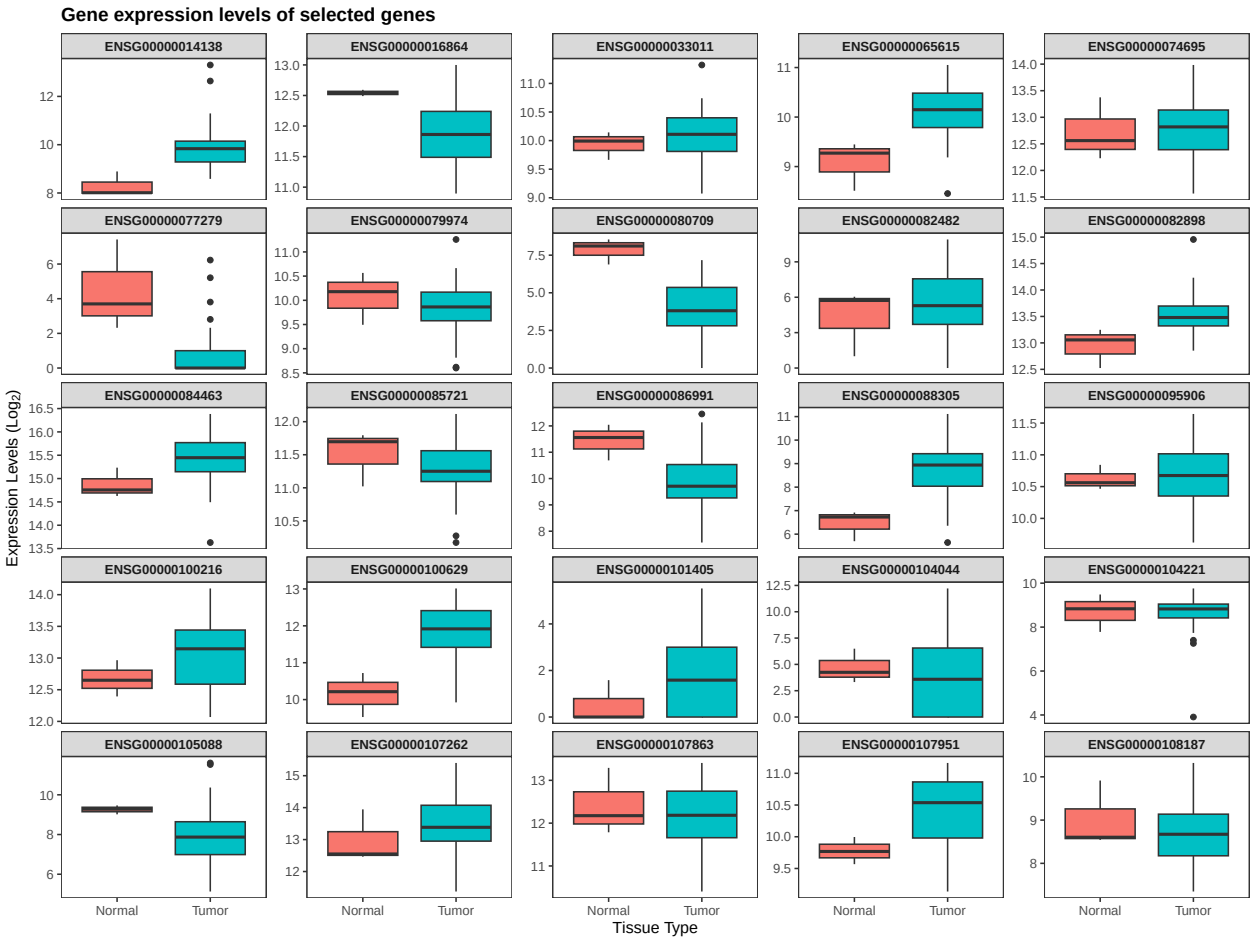
```
## ENSG00000000005                           0                           13
## ENSG00000000419                        2984                         2484
## ENSG00000000457                        1337                          501
## ENSG00000000460                        1221                           62
## ENSG00000000938                         215                          641
##                 TCGA.FU.A3EO.11A.13R.A213.07 TCGA.HM.A3JJ.11A.12R.A21T.07
## ENSG00000000003                        2299                         1317
## ENSG00000000005                          17                           46
## ENSG00000000419                        2157                         2101
## ENSG00000000457                         647                          415
## ENSG00000000460                         149                          164
## ENSG00000000938                         448                          909
```

```
names(counts_data) <- gsub(x=names(counts_data),
                           pattern = '\\.',
                           replacement = '-')
```

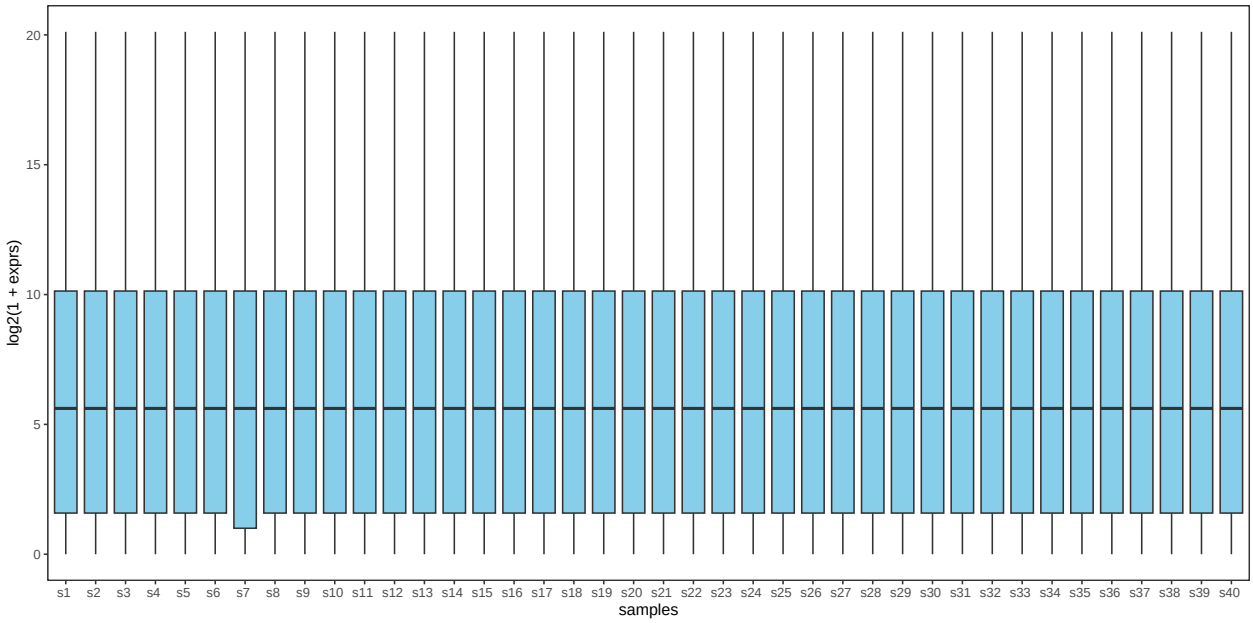
Data Visualisation

```
# randomly selecting statistically significant genes with FDR < 1% & |logFC| > 2
set.seed(10)
selected_ids <- sample(1:nrow(filter(dea_results, FDR < 0.05, abs(logFC) > 2)), 25)
selected_genes <- rownames(counts_data)[selected_ids]
```

```
t(counts_data[selected_genes, ]) |>
  as.data.frame() |>
  rownames_to_column(var='ID') |>
  dplyr::inner_join(metadata, by=c('ID' = 'sampleIDs')) |>
  mutate(group = factor(group)) |>
  pivot_longer(cols=c(-ID,-group), values_to = 'expr', names_to = 'genes') |>
  select(-ID) |>
  mutate(expr = log2(1+expr)) |>
  ggplot(aes(group, expr, fill=group)) +
  geom_boxplot() +
  facet_wrap(~genes, nrow = 5, ncol = 5, scales='free_y') +
  theme_bw() +
  theme(legend.position = 'none',
        panel.grid = element_blank(),
        plot.title = element_text(face='bold'),
        strip.text = element_text(face='bold')) +
  labs(title= 'Gene expression levels of selected genes', x='Tissue Type',
       y=expression('Expression Levels '*'(Log'[2]*')'))
```




```
counts_data |>
  rownames_to_column(var='ID') |>
  pivot_longer(cols=-ID, names_to = 'samples', values_to = 'exprs') |>
  mutate(samples = str_extract(samples, 'TCGA-\\w{2}-\\w{4}')) |>
  ggplot(aes(samples, log2(1+exprs))) +
  geom_boxplot(fill='skyblue') +
  theme_bw() +
  theme(panel.grid = element_blank(),
        plot.title = element_text(face='bold')) +
  scale_x_discrete(labels=paste0('s', seq(1,40), sep=' '))
```




```
# summary statistics of genes (log2 expression values)
counts_data |>
  rownames_to_column(var='ID') |>
  pivot_longer(cols=-ID, names_to = 'samples', values_to = 'exprs') |>
  mutate(exprs = log2(1+exprs)) |>
  summarize(mean_val = mean(exprs),
            median_val = median(exprs),
            sd_val = sd(exprs),
            .by = ID) |>
  arrange(desc(mean_val)) |>
  head(10)
```

```
## # A tibble: 10 x 4
##    ID              mean_val median_val sd_val
##    <chr>              <dbl>      <dbl>  <dbl>
##  1 ENSG00000198804     19.0       19.2  1.01
##  2 ENSG00000198938     18.6       18.7  0.873
##  3 ENSG00000198886     18.1       18.2  0.984
##  4 ENSG00000135046     18.1       18.6  1.72
##  5 ENSG00000198712     18.0       18.3  0.812
##  6 ENSG00000092199     17.7       17.9  0.618
##  7 ENSG00000156508     17.7       17.6  0.704
##  8 ENSG00000145741     17.7       17.4  0.853
##  9 ENSG00000210082     17.7       17.6  0.938
## 10 ENSG00000198727     17.5       17.8  1.08
```

Preprocessing for ML

Feature Selection

- Remove statistically not significant genes

```r
remove_genes <- dea_results |>
  filter(FDR >= 0.05) |>
  rownames()
```

```r
# droppping statistically insignificant genes
counts_data <- counts_data[-which(remove_genes %in% rownames(counts_data)),]
```

```r
# transpose for ML (merging target class)
prep_data <- t(counts_data) |>
  as.data.frame() |>
  rownames_to_column('sampleIDs') |>
  dplyr::inner_join(metadata, by='sampleIDs') |>
  mutate(group = factor(group),
         group = relevel(group, ref='Normal')) |>
  column_to_rownames('sampleIDs')
```

```r
# split into train and test data
set.seed(20)

split_data <- function(df, p=0.75){
  # get majority and minority class
  majority_class <- prep_data |>
  filter(group == 'Tumor')

  minority_class <- prep_data |>
  filter(group == 'Normal')

  # split majority and minority class into train and test data
  majority_train <- majority_class |>
  slice_sample(n=as.integer(p*nrow(majority_class)))

  majority_test <- majority_class |>
    filter(!(rownames(majority_class)  %in% rownames(majority_train)))

  # sample 2 from minority class
  minority_train <- minority_class |>
  slice_sample(n=2)

  minority_test <- minority_class |>
    filter(!(rownames(minority_class)  %in% rownames(minority_train)))


  train <- bind_rows(majority_train, minority_train)
  test <- bind_rows(majority_test, minority_test)

  return(list(train=train, test=test))
}

train_test_data <- split_data(prep_data)
```

```r
train_data <- train_test_data$train
test_data <- train_test_data$test
```

```
xtrain <- train_data %>% select(-group)
xtest <- test_data %>% select(-group)
ytrain <- train_data$group
ytest <- test_data$group
```

- Feature selection 2: We will fit a Lasso regression model to remove features that are shurnk to zero

To do this, we will first transform the expression levels, normalise and fit a logistic regression model

We will do this by creating a pipeline

```
scale_preprocessor <- function(df_train, df_test=NULL){
  df_train_log <- log2(1+df_train)

  mean_vals <- rowMeans(as.matrix(t(df_train_log)))
  sd_vals <- matrixStats::rowSds(as.matrix(t(df_train_log)))

  df_train_log <- (t(df_train_log) - mean_vals) / sd_vals

  if (!is.null(df_test)){
    df_test_log <- log2(1+df_test)
    df_test_log <- (t(df_test_log) - mean_vals) / sd_vals

    return(list(as.data.frame(t(df_train_log)),
                as.data.frame(t(df_test_log))
                )
           )
  } else{
    return (as.data.frame(t(df_train_log)))
  }
}
```

```
scaled_data <- scale_preprocessor(xtrain, xtest)

scaled_train <- scaled_data[[1]]
scaled_test <- scaled_data[[2]]
```

```
# check for genes with missing values as a result of scaling
# 5 variables have missing values in them
which(scaled_train |>
  apply(2, function(x) sum(is.na(x))) > 0)
```

```
## ENSG00000268799 ENSG00000269138 ENSG00000273177 ENSG00000275508 ENSG00000279273
##              15              61             902            1323            1922
```

```
remove_na_variables <- c("ENSG00000268799", "ENSG00000269138",
                         "ENSG00000273177", "ENSG00000275508",
                         "ENSG00000279273")
```

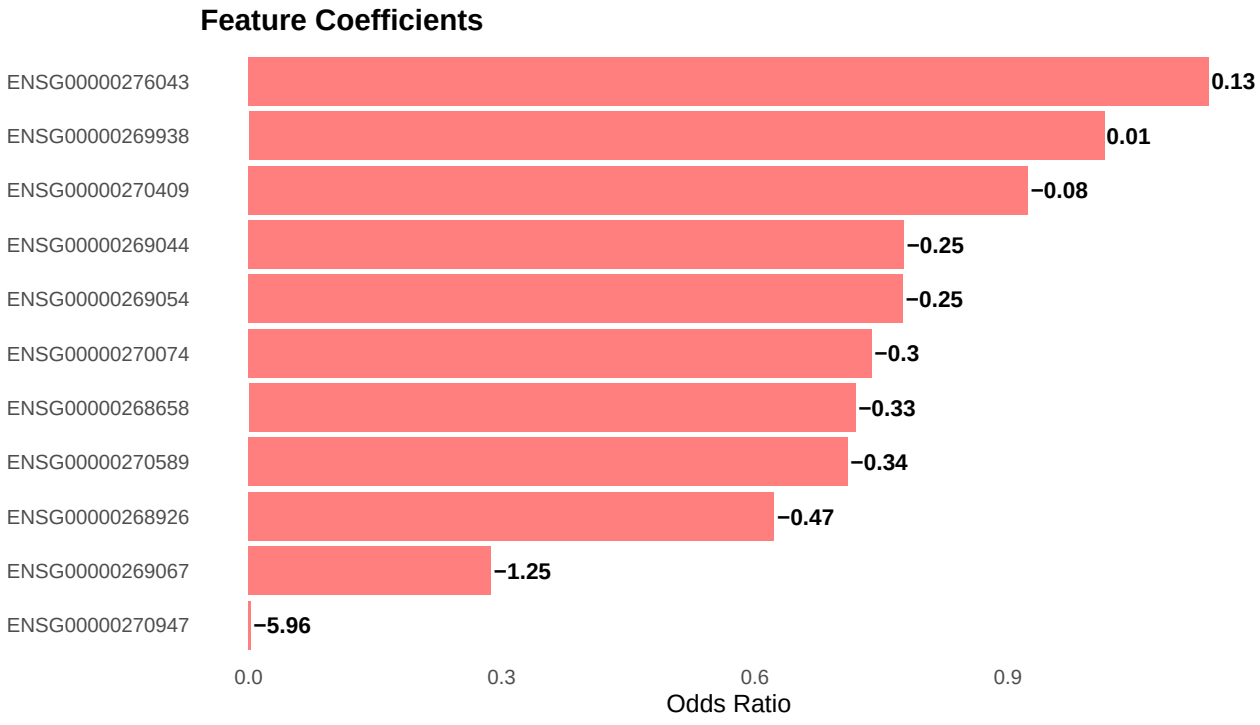
```r
# remove genes with all missing values
scaled_train <- select(scaled_train, -all_of(remove_na_variables))
scaled_test <- select(scaled_test, -all_of(remove_na_variables))
```

```r
# fit a lasso logistic regression model
# alpha = 1 (Lasso)
lasso <- glmnet(as.matrix(scaled_train), ytrain, lambda =2e-7,
                family=binomial, alpha=1, standardize = F)
```

```r
# predictions on train
lasso_probabilities <- predict(lasso, as.matrix(scaled_train), type = 'response')
lasso_predictions <- factor(ifelse(lasso_probabilities > 0.5, 'Tumor', 'Normal'))
lasso_predictions <- relevel(lasso_predictions, ref='Normal')
```

```r
# select coefficients with non-zero values
selected_features <- as.data.frame(as.matrix(coefficients(lasso))) |>
  arrange(s0) |>
  filter(s0 != 0) |>
  rownames_to_column('features') |>
  mutate(odds_ratio = exp(s0)) |>
  filter(features != '(Intercept)')

selected_features %>%
  ggplot(aes(y=reorder(features, odds_ratio), x=odds_ratio)) +
  geom_col(fill='red', alpha=0.5) +
  geom_text(aes(label=round(s0,2)), hjust = -.05, size=3.4, fontface='bold') +
  theme_minimal() +
  theme(panel.grid = element_blank(),
        plot.title = element_text(face='bold')) +
  labs(title = 'Feature Coefficients', x='Odds Ratio', y='')
```

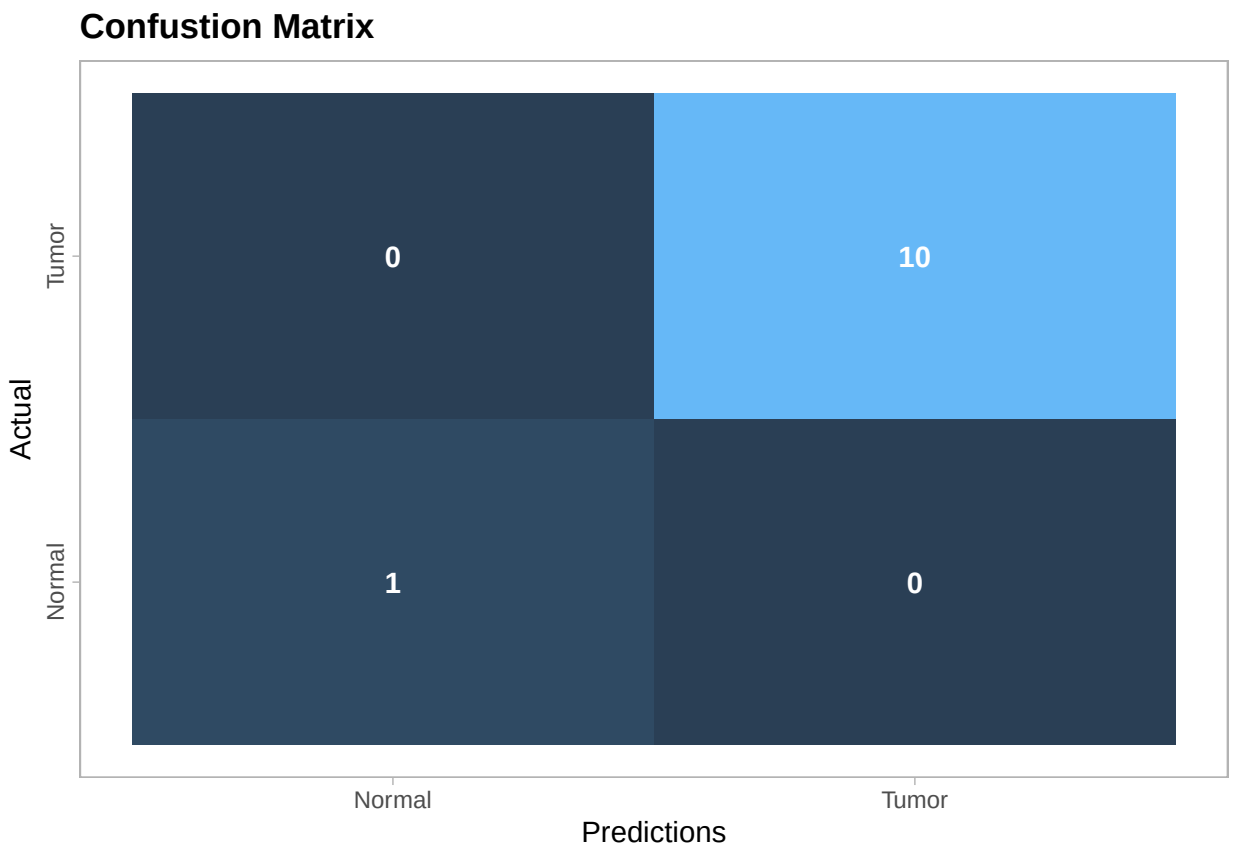


Building a kNN model

```
# kNN model
knn_model <- train.kknn(group ~ ., data = cbind(
  select(scaled_train, all_of(selected_features$features)),
  group=ytrain), kmax = 5)
```

```
knn_predictions <- predict(knn_model, scaled_test)
```

```
as.data.frame(table(Predictions=knn_predictions, Actual=ytest)) |>
  ggplot(aes(y=Actual, x=Predictions, fill=Freq)) +
  geom_tile(alpha=0.9) +
  geom_text(aes(label=Freq), color='white', fontface='bold') +
  theme_light() +
  theme(panel.grid = element_blank(),
        legend.position = 'none',
        axis.text.y = element_text(angle=90, hjust=0.5),
        plot.title = element_text(face='bold')) +
  labs(title = 'Confustion Matrix', y='Actual', x='Predictions')
```




```
model_performance <- function(model, X, y){
  model_predictions <- predict(model, X)
  acc <- mean(y == model_predictions)
  recall <- MLmetrics::Recall(y, model_predictions, positive = 'Tumor')
```

```
  f1 <- MLmetrics::F1_Score(y, model_predictions, positive = 'Tumor')
  specificity <- MLmetrics::Specificity(y, model_predictions, positive = 'Tumor')

  res <- t(data.frame(Accuracy = acc, Recall = recall, F1=f1, Specificity=specificity))
  data.frame(scores=100*res)
}


print(model_performance(knn_model, scaled_test, ytest))
```

```
##             scores
## Accuracy       100
## Recall         100
## F1             100
## Specificity    100
```

```
print(model_performance(knn_model, scaled_test, ytest))
```

```
##             scores
## Accuracy       100
## Recall         100
## F1             100
## Specificity    100
```