Models for discussion in report

Timothy Hedspeth

```
# This Rmarkdown will be the primary file in terms of modeling with for our
# project, this file contains the model fitting for our 4 models of interest
# risk score models and classification trees for Salmonella and E. Coli


##~~~~~~~~~~~~~~~~~~~~~~~~~~##
#### Data and packages ####
##~~~~~~~~~~~~~~~~~~~~~~~~~~##


rm(list=ls())

library(kableExtra)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.0     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
```

```
library(ggpubr)
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
```

```
##
##     expand, pack, unpack

## Loaded glmnet 4.1-4
```

```
library(bestglm)
```

```
## Loading required package: leaps
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(DescTools)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE

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

```
library(kableExtra)
library(ggpubr)
library(rpart)
library(rpart.plot)

setwd("~/Desktop/Semester_3/Practical/Final") # Where all our data is
salmonella <- read.csv("final_salmonella.csv")
ecoli <- read.csv("final_ecoli.csv")

# Functions required for this analysis (available on our Github)
source("functions_outbreaks_and_lasso.R")
source("compress_levels.R")
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

##
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':
##
```

```
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, splice
```

Section 1; Salmonella

We decided to first examine the bacteria Salmonella. Recall that our interest lays in predicting if a case is part of an outbreak, in the case we define an outbreak to be **8 or more cases** from the same SNP.cluster in a given month and year. We first remove all NA values, as we discuss in our EDA that the degree of missingness does not allow for us to impute the data and trust the results. After defining an outbreak we compress the levels of the SNP.cluster to be the top 7 most observed across our time frame, and create our test train splits. In other versions of this work we included interaction terms though the results were found to not be as interpretable as our additive models, an important aspect as we do hope that our models could be used as tools for individuals in the public health sphere that do not have extensive statistical training. All interpretation of these models/metrics will be in our final report which we will make avilable in our repository.

```r
#~~~~~~~~~~~~~~~~~~#
## Clean the data ##
#~~~~~~~~~~~~~~~~~~#

# Complete case analysis
salmonella <- salmonella %>% filter(!is.na(month) & !is.na(year) &
                                      !is.na(Min.same)  & !is.na(Serovar) &
                                      !is.na(Isolation.type)  & !is.na(AMR.genotypes) &
                                      !is.na(Computed.types) & !is.na(SNP.cluster)) %>%
                              dplyr::select(-c(Min.diff, Location))

# We will add an outbreak to the model
# 8 cases = outbreak
salmonella <- add_outbreak(salmonella, 8)
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the
## `.groups` argument.
```

```r
# Compress the levels of SNP clusters to be the top overall
salmonella$SNP.cluster <- compress_levels_factors(salmonella, 'SNP.cluster')


## Create a test and train split
salmonella_train <- salmonella %>% filter(year != 2021 & year !=  2019 &
                                          year != 2020 & year !=  2022)
salmonella_test <- salmonella  %>%  filter(year == 2021 | year ==  2019 )
# table(salmonella_test$outbreak)
#table(salmonella_train$outbreak)
Percent_outbreak_train <- 2084/(2084+11559)
Percent_outbreak_test <- 332/(332+2074)

# Get rid of variables that we won't use in analysis
salmonella_train <- salmonella_train %>% dplyr::select(-c(year, Min.same))
salmonella_test <- salmonella_test %>% dplyr::select(-c(year, Min.same))

# Make the variables factors
salmonella_train[] <- lapply(salmonella_train, function(x){return(as.factor(x))})
salmonella_test[] <- lapply(salmonella_test, function(x){return(as.factor(x))})
```

We will first fit our Lasso Regularized Logistic Regression as we include many factor variables that all have 8 or more levels to predict if a case is part of an outbreak using biological, spatial and time of the year. Lasso aided in our model selection procedure as coefficients are snapped to 0 when they are weakly associated with the outcome. These coefficients were divided by the median non-zero coefficients and rounded to get the corresponding risk score coefficients.

```r
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##
#### Salmonella, Risk Score ####
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~##

# Perform Lasso
salmonella_coef <- lasso(salmonella_train, 12)
salmonella_coef <- as.data.frame(as.matrix(salmonella_coef))
sal_coef <- as.vector(salmonella_coef$s1)

# Divide the Lasso
salmonella_coefs  <- round(salmonella_coef$s1/median(sal_coef[sal_coef != 0]))
salmonella_coef$s1 <- salmonella_coefs


## Some coefficients that we trained the data on will not be observed in the
## test set so we will need to add these to be able to assess how the
## model perform on testing data


# We note that the factors relating to SNP and AMR are the issue as
# some of the levels are not observed, so we will add them now so the
# score can be calculated

# SNP cluster
levels_snp_train <- as.list(levels(salmonella_train$SNP.cluster))
salmonella_test$SNP.cluster <- factor(salmonella_test$SNP.cluster,
                                      levels = levels_snp_train)
salmonella_test$SNP.cluster <- relevel(salmonella_test$SNP.cluster, "Other")

# AMR genotypes
levels_AMR_train <- as.list(levels(salmonella_train$AMR.genotypes))
salmonella_test$AMR.genotypes <- factor(salmonella_test$AMR.genotypes,
                                        levels = levels_AMR_train)
salmonella_test$AMR.genotypes <- relevel(salmonella_test$AMR.genotypes, "aadA1=COMPLETE,gyrA_D87Y=POINT


# We need to make sure that all levels of the factor are accounted for
#dim(model.matrix(outbreak~., salmonella_test))
#dim(model.matrix(outbreak~., salmonella_train))


# All checks out!




## Create and display a table of Non-Zero coefficients

sal_coefs_table <- salmonella_coef %>%
```

```r
                   filter(s1 != 0) %>%
                   dplyr::rename(Score =  s1)

sal_coefs_table %>%  kbl(caption = "Score coefficents for Salmonella",
                        booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

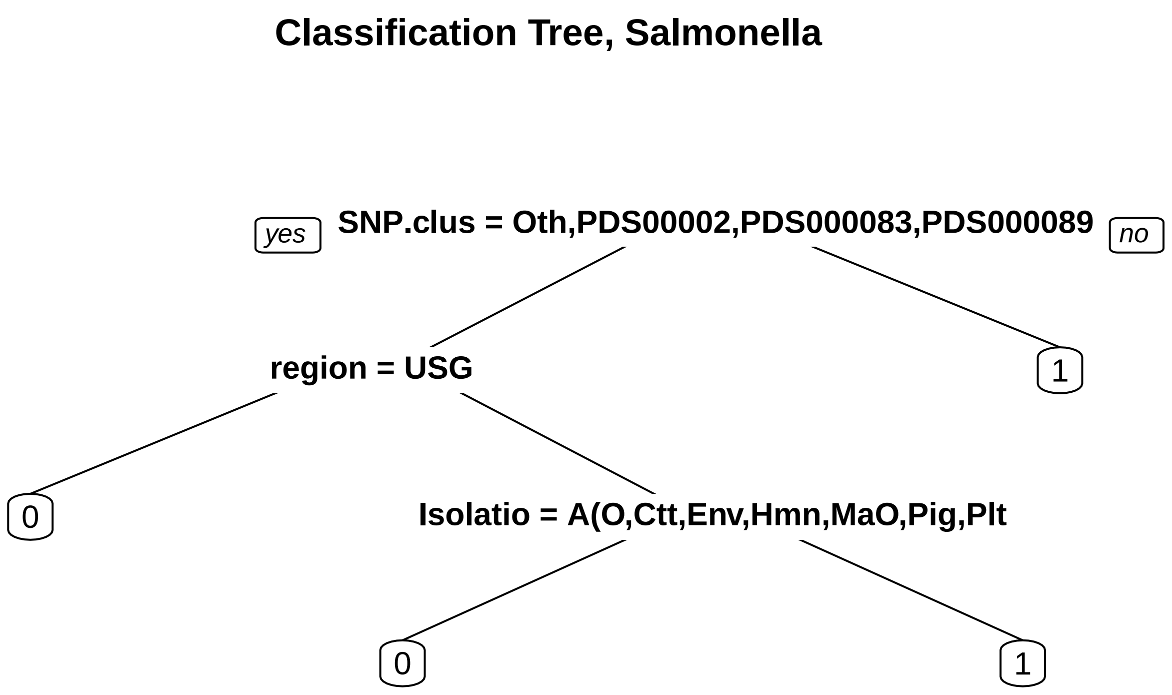
Table 1: Score coefficents for Salmonella

	Score
SNP.clusterPDS000030237.975	3
SNP.clusterPDS000032668.817	1
SNP.clusterPDS000120941.3	1
Computed.typesantigen_formula=9:g,m:-,serotype=Enteritidis	1
Isolation.source.categoryProduce and Food	1
regionUSA, General	-2

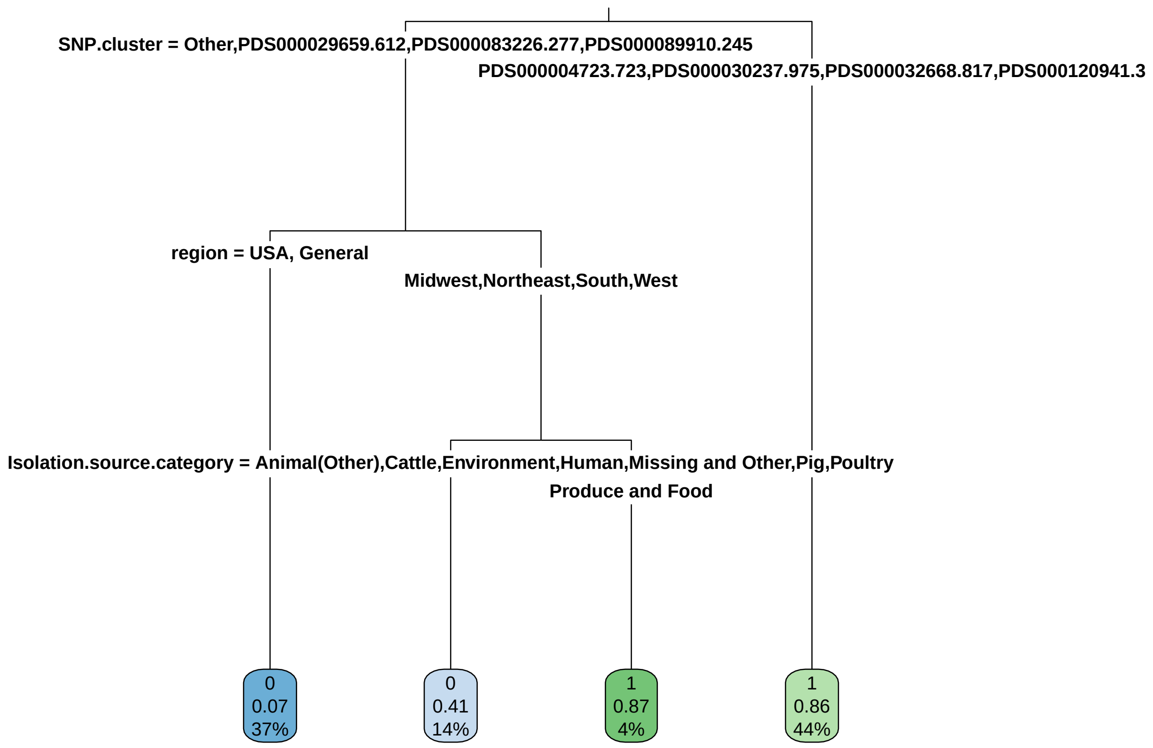
Tree

Code for fitting the tree model:

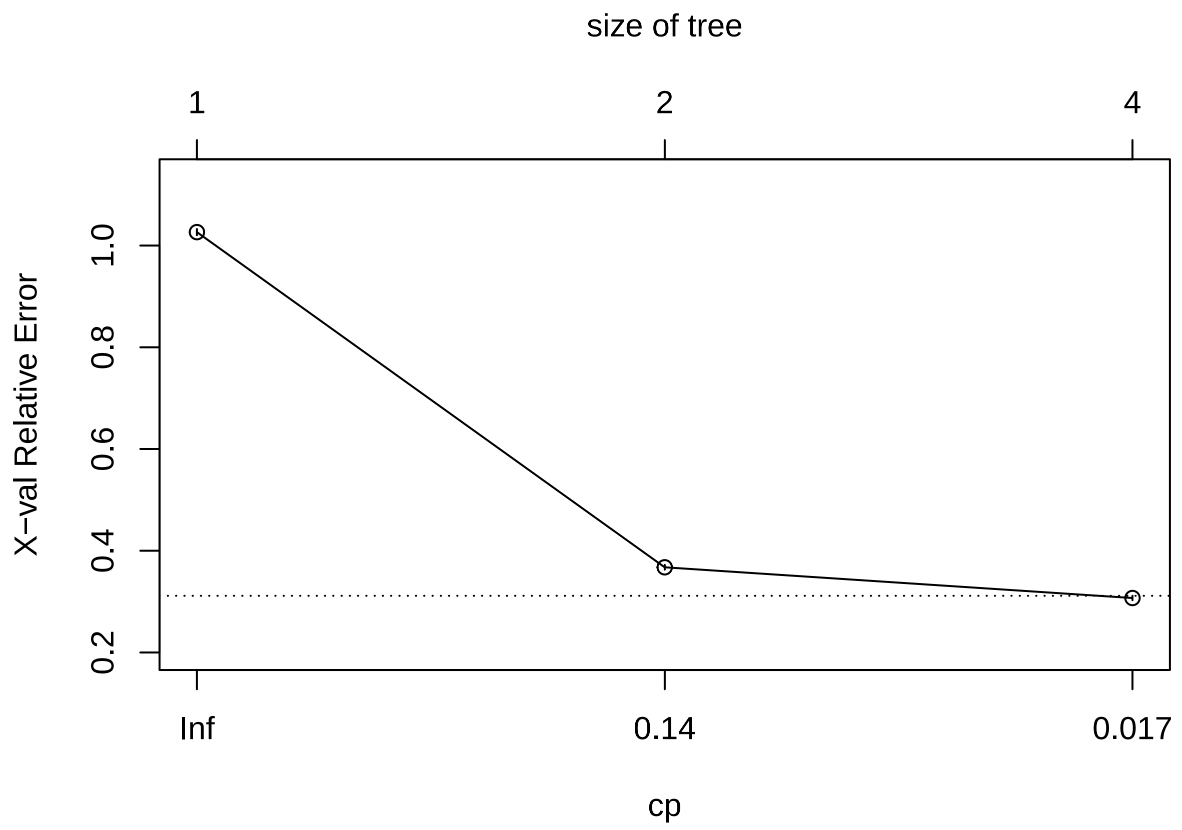
```r
##~~~~~~~~~~~~~~~~~~~~~~##
#### Salmonella Tree ####
##~~~~~~~~~~~~~~~~~~~~~~##


# We need to weight the tree (its done automatically in lasso)
prop_zero <- 1/(sum(salmonella_train$outbreak == 0)/nrow(salmonella_train))
prop_one <- 1/(sum(salmonella_train$outbreak == 1)/nrow(salmonella_train))

sal_weights <- ifelse(salmonella_train$outbreak == 0, prop_zero, prop_one)


# Fit the tree
tree_salmonella <- rpart(outbreak ~ ., data = salmonella_train,
                         method = "class", weights = sal_weights,
                         control=rpart.control(minsplit = 20,
                                               minbucket = 30))

# Plot the tree
prp(tree_salmonella,
    main = "Classification Tree, Salmonella")
```


Classification Tree, Salmonella
yes
SNP.clus = Oth,PDS00002,PDS000083,PDS000089
no
region = USG
1
0
Isolatio = A(O,Ctt,Env,Hmn,MaO,Pig,Plt
0
1

```
# This plot has explicit splits but the text is to difficult to read
rpart.plot(tree_salmonella,
           cex =.6,
           type = 3)
```


SNP.cluster = Other,PDS000029659.612,PDS000083226.277,PDS000089910.245
PDS000004723.723,PDS000030237.975,PDS000032668.817,PDS000120941.3
region = USA, General
Midwest,Northeast,South,West
Isolation.source.category = Animal(Other),Cattle,Environment,Human,Missing and Other,Pig,Poultry
Produce and Food
0 0.07 37%
0 0.41 14%
1 0.87 4%
1 0.86 44%

```
# Looking  at complexity
plotcp(tree_salmonella)
```



```
# The tree already is meeting the standards of minimizing complexity
# we will not further prune it
```

Diagnostics

We will begin with diagnostics regarding the Risk score models:

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
## Diagnostics; salmonella  ##
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~#


# Get the risk scores in the testing data
variables_sal <- model.matrix(outbreak~., salmonella_test)
salmonella_test$score <- variables_sal[,-1] %*% salmonella_coefs[-1]

# Fit a regression to see how it does
mod_salmonella <- glm(outbreak ~ score, data=salmonella_test, family = quasibinomial())

# For confusion matrices
salmonella_test$predicted <- predict(mod_salmonella, type = "response")

# ROC
risk_perfomance_salmonella <- roc(outbreak ~ predicted, data=salmonella_test)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
## How does the scoring system do?

# Test
salmonella_test$score <- as.numeric(salmonella_test[,10])
salmonella_test$outbreak_1 <- case_when(salmonella_test$outbreak == "0" ~ 0,
                                        salmonella_test$outbreak == "1" ~ 1)
salmonella_test  %>% group_by(score) %>%
                     dplyr::summarize(pct = 100*round(sum(outbreak_1)/n(),4)) %>%
                     dplyr::rename("Percent outbreaks" = pct) %>%
                     kbl(caption = "Percent of cases at each score that belong to an outbreak",
                         booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

Table 2: Percent of cases at each score that belong to an outbreak

score	Percent outbreaks
-2	0.00
0	26.41
1	0.00
2	0.00
3	0.00
4	0.00

```r
# Train
sal_train <- model.matrix(outbreak~., salmonella_train)
salmonella_train$scores <- sal_train[,-1] %*% salmonella_coefs[-1]
salmonella_train$score <- as.numeric(salmonella_train[,10])
salmonella_train$outbreak_1 <- case_when(salmonella_train$outbreak == "0" ~ 0,
                                         salmonella_train$outbreak == "1" ~ 1)

salmonella_train %>% group_by(score) %>%
                     dplyr::summarize(pct = 100*round(sum(outbreak_1)/n(),4)) %>%
                     dplyr::rename("Percent outbreaks" = pct) %>%
                     kbl(caption = "Percent of cases at each score that belong to an outbreak",
                         booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

Table 3: Percent of cases at each score that belong to an outbreak

score	Percent outbreaks
-2	0.92
-1	9.19
0	15.14
1	31.63
2	58.02
3	80.00
4	91.43
5	100.00

```r
# We want to look at how the models perform on Test and train data  (mainly
# for testing)

# Training data metrics; Salmonella #

# Perform regression and see how the scores do in training
salmonella_train <- salmonella_train[,-c(10,12)]
mod_sal_train <- glm(outbreak ~  score, data = salmonella_train,
                     family =quasibinomial())

salmonella_train$predicted <- predict(mod_sal_train, type ="response")

threshhold <- .5

# ROC
risk_perfomance_salmonella_train <- roc(outbreak ~ predicted,
                                        data=salmonella_train)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
# Get some Coefficents
confusionMatrix(as.factor(ifelse(salmonella_train$predicted > threshhold,
                                 1,0)), salmonella_train$outbreak)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 11089   894
##          1   470  1190
##
##                Accuracy : 0.9
##                  95% CI : (0.8949, 0.905)
##     No Information Rate : 0.8472
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5786
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9593
##             Specificity : 0.5710
##          Pos Pred Value : 0.9254
##          Neg Pred Value : 0.7169
##              Prevalence : 0.8472
##          Detection Rate : 0.8128
##    Detection Prevalence : 0.8783
##       Balanced Accuracy : 0.7652
##
##        'Positive' Class : 0
##
```

```r
# Create a table
metrics_train <- data.frame()
```

```
metrics_train[1,1] <- "AUC"
metrics_train[2,1] <- "Brier Score"
metrics_train[3,1] <- "Accuracy"
metrics_train[4,1] <- "Sensitivity"
metrics_train[5,1] <- "Specificity"

# Fill in  for this training data
metrics_train[1,2] <- round(auc(risk_perfomance_salmonella_train),2)
metrics_train[2,2] <- round(BrierScore(mod_sal_train), 2)
metrics_train[3,2] <- .90
metrics_train[4,2] <- .96
metrics_train[5,2] <- .57



## Metrics for test data; Salmonella ##

confusionMatrix(as.factor(ifelse(salmonella_test$predicted > threshhold,
                                 1,0)), salmonella_test$outbreak)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1068  332
##          1    6    0
##
##                Accuracy : 0.7596
##                  95% CI : (0.7364, 0.7817)
##     No Information Rate : 0.7639
##     P-Value [Acc > NIR] : 0.6601
##
##                   Kappa : -0.0085
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9944
##             Specificity : 0.0000
##          Pos Pred Value : 0.7629
##          Neg Pred Value : 0.0000
##              Prevalence : 0.7639
##          Detection Rate : 0.7596
##    Detection Prevalence : 0.9957
##       Balanced Accuracy : 0.4972
##
##        'Positive' Class : 0
##
```

```
# Define a data frame
metrics <- data.frame()
metrics[1,1] <- "AUC"
metrics[2,1] <- "Brier Score"
metrics[3,1] <- "Accuracy"
metrics[4,1] <- "Sensitivity"
```

```r
metrics[5,1] <- "Specificity"

# Put in Salmonella metrics
metrics[1,2] <- round(auc(risk_perfomance_salmonella),2)
metrics[2,2] <- round(BrierScore(mod_salmonella),2)
metrics[3,2] <- .76
metrics[4,2] <- .99
metrics[5,2] <- 0
```

We will now move to evaluating the tree based model:

```r
### Tree ###

# Predicted on training data
preds = predict(tree_salmonella, salmonella_train, type = "class")

# Predict with Tree
predict_salmonella_tree <- as.data.frame(predict(tree_salmonella, salmonella_train))
names(predict_salmonella_tree) <- c("prob_tree")
salmonella_append <- cbind(salmonella_train, predict_salmonella_tree)

# ROC Analysis
salmonella_acc <- roc(outbreak ~ prob_tree, data =salmonella_append)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls > cases
```

```r
#AUC
salmonella_AUC = round(auc(salmonella_acc), 2)

# Brier Score
salmonella_Brscr = round(sum((predict_salmonella_tree[,2] - (as.numeric(salmonella_train$outbreak) - 1)


## Predict on Test Set ##

pred_test = predict(tree_salmonella, newdata = salmonella_test, type = "class")

# Predict with Tree
predict_test_tree <- as.data.frame(predict(tree_salmonella, salmonella_test))
names(predict_test_tree) <- c("prob_tree")
salmonella_test_append <- cbind(salmonella_test, predict_test_tree)

# ROC Analysis
salmonella_test_acc <- roc(outbreak ~ prob_tree, data =salmonella_test_append)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
#AUC
salmonella_test_AUC = round(auc(salmonella_test_acc), 2)

# Brier Score
salmonella_test_Brscr = round(sum((predict_test_tree[,2] - (as.numeric(salmonella_test$outbreak) - 1))^
# confusion matrix for accuracy, train
conf_mat_sal_train <- table(salmonella_train$outbreak, as.factor(ifelse(predict_salmonella_tree[,2] > .

# Add to the table
metrics_train[1,3] <- salmonella_AUC
metrics_train[2,3] <- salmonella_Brscr
metrics_train[3,3] <- round(sum(diag(conf_mat_sal_train))/sum(conf_mat_sal_train),2)
metrics_train[4,3] <- round(sensitivity(preds, salmonella_train$outbreak),2)
metrics_train[5,3] <- round(specificity(preds, salmonella_train$outbreak),2)


## Metrics for Testing data ##

# confusion matrix for accuracy
conf_mat_sal <- table(salmonella_test$outbreak, as.factor(ifelse(predict_test_tree[,2] > .5, 1,0)))

# Put the values in the table for the training data
metrics[1,3] <- salmonella_test_AUC
metrics[2,3] <- salmonella_test_Brscr
metrics[3,3] <- round(sum(diag(conf_mat_sal))/sum(conf_mat_sal),2)
metrics[4,3] <- round(sensitivity(pred_test, salmonella_test$outbreak),2)
metrics[5,3] <- round(specificity(pred_test, salmonella_test$outbreak),2)
```

E. coli

For E. coli we define an outbreak to be 10 cases. We clean the data below:

```r
##~~~~~~~~~~~~~~~~~~~~~~~~~~##
#### E. Coli Risk Scores ####
##~~~~~~~~~~~~~~~~~~~~~~~~~~##

# Read in the ecoli data
ecoli <- read.csv("final_ecoli.csv")


## We will do a complete case analysis ##
ecoli <- ecoli %>% filter(!is.na(month) & !is.na(year) &
                            !is.na(SNP.cluster)  & !is.na(Strain) &
                            !is.na(AMR.genotypes) & !is.na(Location)&
                            !is.na(Isolation.type) & !is.na(region) &
                            !is.na(Isolation.source.category)) %>%
                    dplyr::select(-c(day, Min.same, Location))




# We will add an outbreak to the model
ecoli <- add_outbreak(ecoli, 10)
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the
## `.groups` argument.
```

```r
# Compress the SNP Cluster levels
ecoli$SNP.cluster <- compress_levels_factors(ecoli, 'SNP.cluster')
```

```r
## Get test and train splits ##
ecoli_train <- ecoli %>% filter(year != 2021 & year !=  2019 )
ecoli_test <- ecoli  %>%  filter(year == 2021 | year ==  2019 |
                                 year == 2020 | year ==  2022)
#table(ecoli_train$outbreak)
Percent_outbreak_test_ecoli <- 175/(175+1811)
Percent_outbreak_train_ecoli <- 842/(842+4710)


ecoli_train <- ecoli_train %>% dplyr::select(-c(year))
ecoli_test <- ecoli_test %>% dplyr::select(-c(year))


ecoli_train[] <- lapply(ecoli_train, function(x){return(as.factor(x))})
ecoli_test[] <- lapply(ecoli_test, function(x){return(as.factor(x))})
```

Fit the model:

```r
# Lasso
ecoli_coef <- lasso(ecoli_train, 12)
ecoli_coef <- as.data.frame(as.matrix(ecoli_coef))
eco_coef <- as.vector(ecoli_coef$s1)
ecoli_coefs  <- round(ecoli_coef$s1/median(eco_coef[eco_coef != 0]))
ecoli_coef$s1 <- ecoli_coefs

# Need  to check strain, SNP cluster,


# Make sure all the factors are interpe

#levels(ecoli_test$month_source) == levels(ecoli_train$month_source)
strain_levels_train <- as.list(levels(ecoli_train$Strain))
ecoli_test$Strain <- factor(ecoli_test$Strain,
                            levels = strain_levels_train)
snp_levels_train <- as.list(levels(ecoli_train$SNP.cluster))
ecoli_test$SNP.cluster <- factor(ecoli_test$SNP.cluster,
                                 levels = snp_levels_train)




# Make sure all
dim(model.matrix(outbreak~., ecoli_test))
```

```
## [1] 1986   42
```

```r
dim(model.matrix(outbreak~., ecoli_train))
```

```
## [1] 5552   42
```

```r
# Coefficents
eco_coefs_table <- ecoli_coef %>%
                   filter(s1 != 0) %>%
                   dplyr::rename(Score =  s1)

eco_coefs_table %>%  kbl(caption = "Score coefficents",booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

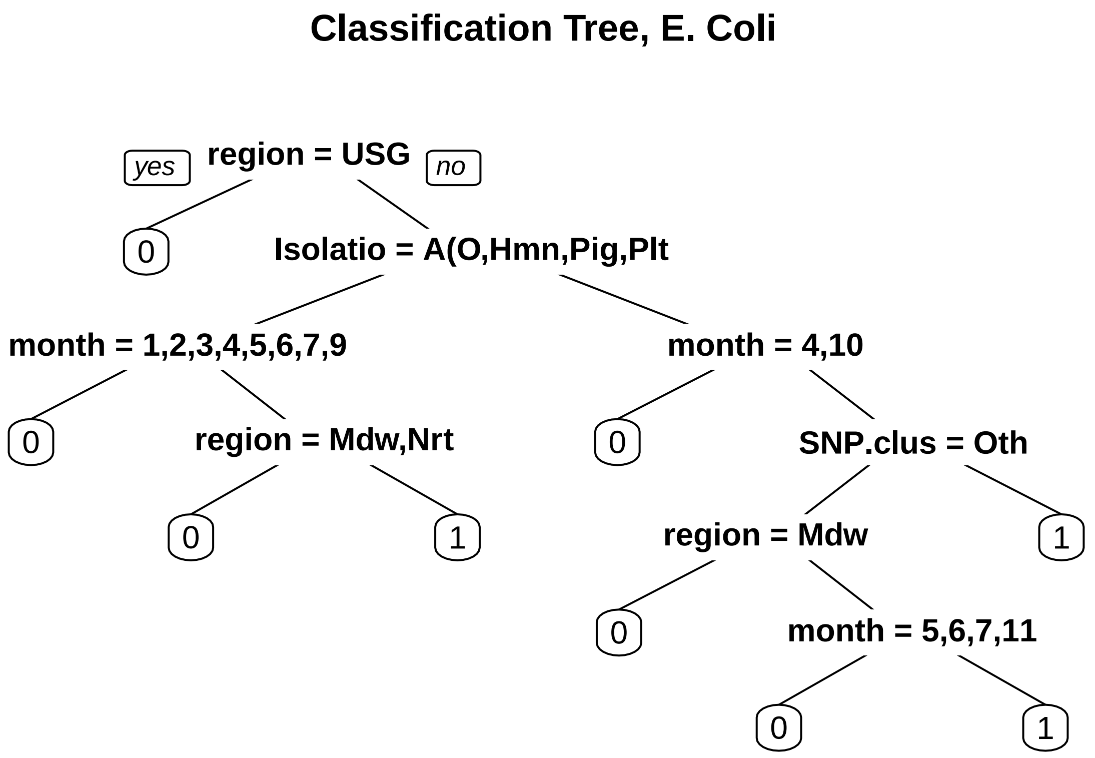
Table 4: Score coefficents

	Score
(Intercept)	1
SNP.clusterPDS000076681.93	2
AMR.genotypesacrF=MISTRANSLATION,blaEC=COMPLETE,mdtM=COMPLETE	2
Isolation.source.categoryHuman	-1
Isolation.source.categoryPoultry	-3
regionNortheast	3
regionUSA, General	-7

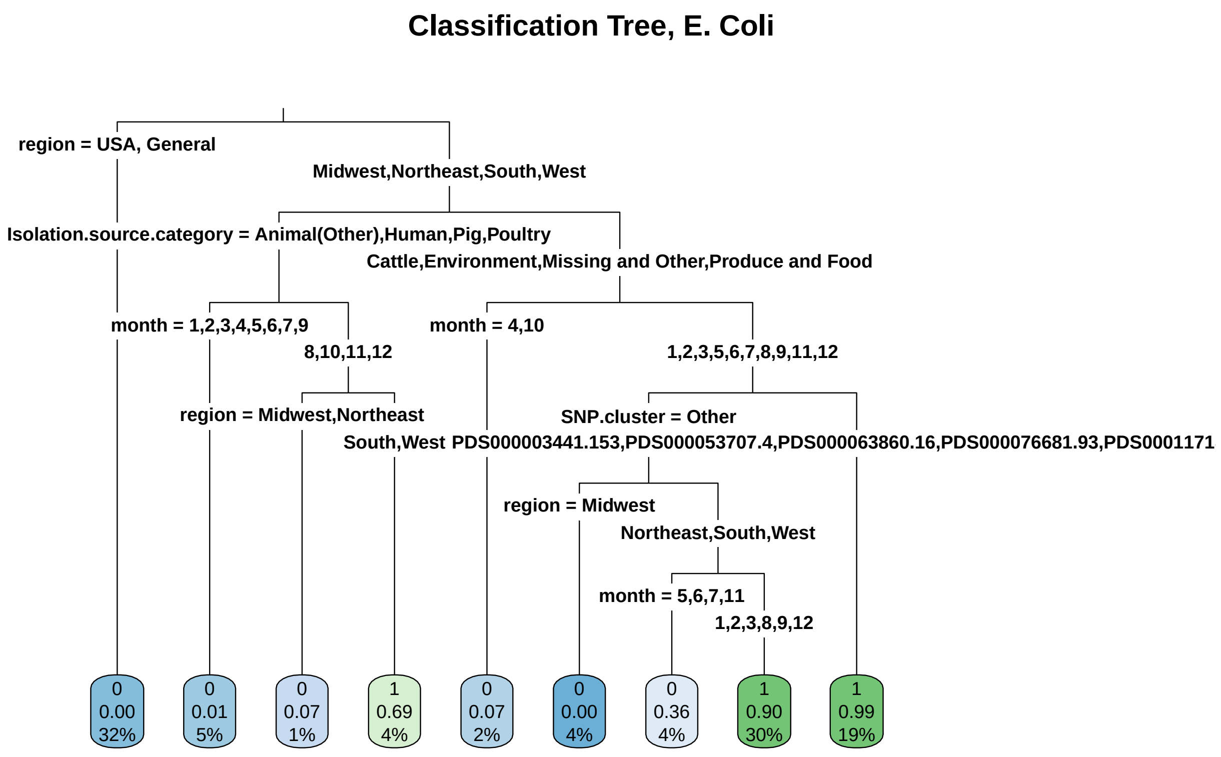
Tree

Fit the tree model:

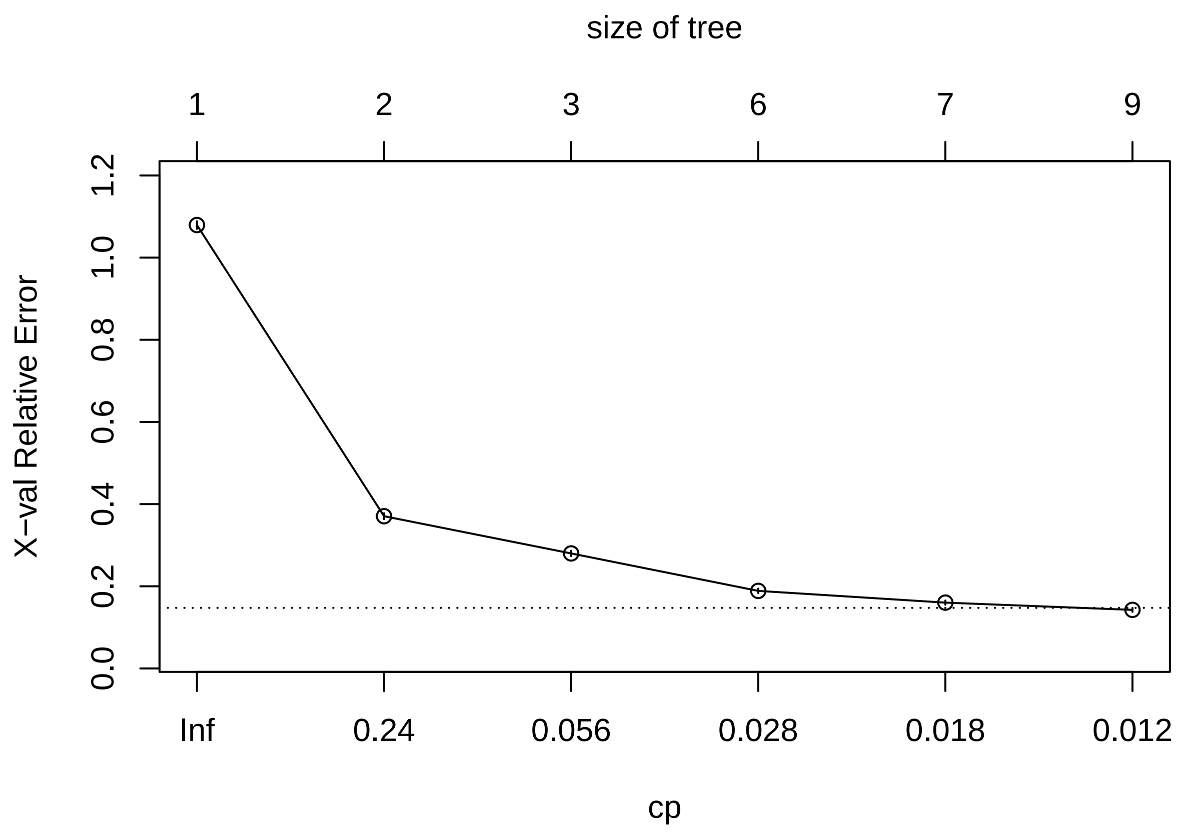
```
# We need to weight the tree (its done automatically in lasso)
prop_zero <- 1/(sum(ecoli_train$outbreak == 0)/nrow(ecoli_train))
prop_one <- 1/(sum(ecoli_train$outbreak == 1)/nrow(ecoli_train))

eco_weights <- ifelse(ecoli_train$outbreak == 0, prop_zero, prop_one)


# Fit the tree
tree_ecoli <- rpart(outbreak ~ ., data = ecoli_train,
                         method = "class", weights = eco_weights,
                         control=rpart.control(minsplit = 20,
                                               minbucket = 30))

# Plot the tree
prp(tree_ecoli,
    main = "Classification Tree, E. Coli")
```



```
# Tree with all splits labeled
rpart.plot(tree_ecoli,
           cex=.6,
           type = 3,
           main = "Classification Tree, E. Coli")
```


Classification Tree, E. Coli
region = USA, General
Midwest,Northeast,South,West
Isolation.source.category = Animal(Other),Human,Pig,Poultry
Cattle,Environment,Missing and Other,Produce and Food
month = 1,2,3,4,5,6,7,9
8,10,11,12
month = 4,10
1,2,3,5,6,7,8,9,11,12
region = Midwest,Northeast
South,West
SNP.cluster = Other
PDS000003441.153,PDS000053707.4,PDS000063860.16,PDS000076681.93,PDS0001171
region = Midwest
Northeast,South,West
month = 5,6,7,11
1,2,3,8,9,12
0 0.00 32%
0 0.01 5%
0 0.07 1%
1 0.69 4%
0 0.07 2%
0 0.00 4%
0 0.36 4%
1 0.90 30%
1 0.99 19%

```r
# Looking  at complexity
plotcp(tree_ecoli)
```



```r
# The tree already is meeting the standards of minimizing complexity
# we will not further prune it
```

Measures

For risk score:

```r
# Get the scores for test set
mat_ecoli_test <- model.matrix(outbreak~., ecoli_test)
ecoli_test$score <- mat_ecoli_test[,-1] %*% ecoli_coefs[-1]

# Perform a logistic regression to see how it performs
mod_ecoli <- glm(outbreak ~ score, data=ecoli_test, family = quasibinomial())


# For confusion matrices
ecoli_test$predicted <- predict(mod_ecoli, type = "response")
risk_perfomance_ecoli <- roc(outbreak ~ predicted, data=ecoli_test)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
confusionMatrix(as.factor(ifelse(ecoli_test$predicted > threshhold,
                                 1,0)), ecoli_test$outbreak)
```

```
## Warning in confusionMatrix.default(as.factor(ifelse(ecoli_test$predicted > :
```

```
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1811  175
##          1    0    0
##
##                Accuracy : 0.9119
##                  95% CI : (0.8985, 0.924)
##     No Information Rate : 0.9119
##     P-Value [Acc > NIR] : 0.5201
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9119
##          Neg Pred Value :    NaN
##              Prevalence : 0.9119
##          Detection Rate : 0.9119
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
# How does the risk score perform?

##  Test
ecoli_test$score <- as.numeric(ecoli_test[,9])
ecoli_test$outbreak_1 <- case_when(ecoli_test$outbreak == "0" ~ 0,
                                   ecoli_test$outbreak == "1" ~ 1)
ecoli_test %>% group_by(score) %>%
                     dplyr::summarize(pct = 100*round(sum(outbreak_1)/n(),4)) %>%
                     dplyr::rename("Percent outbreaks" = pct) %>%
                     kbl(caption = "Percent of cases at each score that belong to an outbreak",
                         booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

```
## Train
mat_ecoli_train <- model.matrix(outbreak~., ecoli_train)
ecoli_train$scores <- mat_ecoli_train[,-1] %*% ecoli_coefs[-1]
ecoli_train$score <- as.numeric(ecoli_train[,9])
ecoli_train$outbreak_1 <- case_when(ecoli_train$outbreak == "0" ~ 0,
                                    ecoli_train$outbreak == "1" ~ 1)

ecoli_train  %>% group_by(score) %>%
                     dplyr::summarize(pct = 100*round(sum(outbreak_1)/n(),4)) %>%
                     dplyr::rename("Percent outbreaks" = pct) %>%
                     kbl(caption = "Percent of cases at each score that belong to an outbreak",
```

Table 5: Percent of cases at each score that belong to an outbreak

score	Percent outbreaks
-8	0.00
-7	0.00
-6	0.00
-3	0.00
-1	0.00
0	18.93
1	92.31
2	21.43
3	0.00

```
                      booktabs=T, escape=T, align = "c") %>%
                  kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

Table 6: Percent of cases at each score that belong to an outbreak

score	Percent outbreaks
-8	0.00
-7	0.36
-6	0.00
-5	0.00
-4	0.00
-3	0.00
-1	7.24
0	13.57
1	47.37
2	80.19
3	76.90
5	0.00

```
# Get metrics for training data
ecoli_train <- ecoli_train[,-c(9,11)]
mod_eco_train <- glm(outbreak ~  score, data = ecoli_train,
                     family =quasibinomial())

ecoli_train$predicted <- predict(mod_eco_train, type ="response")


# ROC
risk_perfomance_ecoli_train <- roc(outbreak ~ predicted,
                                        data=ecoli_train)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
# Get some Coefficents
confusionMatrix(as.factor(ifelse(ecoli_train$predicted > threshhold,
                                 1,0)), ecoli_train$outbreak)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4530  195
##          1  180  647
##
##                Accuracy : 0.9325
##                  95% CI : (0.9255, 0.9389)
##     No Information Rate : 0.8483
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7356
##
##  Mcnemar's Test P-Value : 0.4697
##
##             Sensitivity : 0.9618
##             Specificity : 0.7684
##          Pos Pred Value : 0.9587
##          Neg Pred Value : 0.7823
##              Prevalence : 0.8483
##          Detection Rate : 0.8159
##    Detection Prevalence : 0.8510
##       Balanced Accuracy : 0.8651
##
##        'Positive' Class : 0
##
```

```r
# Add metrics to training data set
metrics_train[1,4] <- round(auc(risk_perfomance_ecoli_train),2)
metrics_train[2,4] <- round(BrierScore(mod_eco_train), 2)
metrics_train[3,4] <- .93
metrics_train[4,4] <- .96
metrics_train[5,4] <- .77


# Add measures to the metrics table for testing
metrics[1,4] <- round(auc(risk_perfomance_ecoli),2)
metrics[2,4] <- round(BrierScore(mod_ecoli),2)
metrics[3,4] <- .91
metrics[4,4] <- 1
metrics[5,4] <- 0
```

For tree model:

```r
## We now want to see how the tree performs ##

## Training set

preds = predict(tree_ecoli, ecoli_train, type = "class")

# Predict with Tree
predict_ecoli_tree <- as.data.frame(predict(tree_ecoli, ecoli_train))
names(predict_ecoli_tree) <- c("prob_tree")
ecoli_append <- cbind(ecoli_train, predict_ecoli_tree)
```

```
# ROC Analysis
ecoli_acc <- roc(outbreak ~ prob_tree, data =ecoli_append)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls > cases
```

```
#AUC
ecoli_AUC = round(auc(ecoli_acc), 2)

# Brier Score
ecoli_Brscr = round(sum((predict_ecoli_tree[,2] - (as.numeric(ecoli_train$outbreak) - 1))^2)/length(eco




## Predict on Test Set ##

pred_test = predict(tree_ecoli, newdata = ecoli_test, type = "class")

# Predict with Tree
predict_test_tree <- as.data.frame(predict(tree_ecoli, ecoli_test))
names(predict_test_tree) <- c("prob_tree")
ecoli_test_append <- cbind(ecoli_test, predict_test_tree)

# ROC Analysis
ecoli_test_acc <- roc(outbreak ~ prob_tree, data =ecoli_test_append)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls > cases
```

```
#AUC
ecoli_test_AUC = round(auc(ecoli_test_acc), 2)

# Brier Score
ecoli_test_Brscr = round(sum((predict_test_tree[,2] - (as.numeric(ecoli_test$outbreak) - 1))^2)/length(



## Metrics tables ##

# Training data

# confusion matrix for accuracy, train
conf_mat_eco_train <- table(ecoli_train$outbreak, as.factor(ifelse(predict_ecoli_tree[,2] > .5, 1,0)))

# Add to the table
metrics_train[1,5] <- ecoli_AUC
metrics_train[2,5] <- ecoli_Brscr
metrics_train[3,5] <- round(sum(diag(conf_mat_eco_train))/sum(conf_mat_eco_train),2)
metrics_train[4,5] <- round(sensitivity(preds, ecoli_train$outbreak),2)
metrics_train[5,5] <- round(specificity(preds, ecoli_train$outbreak),2)


# Testing data
# We want the accuracy of the tree, which we must compute ourseleves
conf_mat_eco <- table(ecoli_test$outbreak, as.factor(ifelse(predict_test_tree[,2] > .5, 1,0)))


# Add measures for the ecoli tree to the metrics table
metrics[1,5] <- ecoli_test_AUC
metrics[2,5] <- ecoli_test_Brscr
metrics[3,5] <- round(sum(diag(conf_mat_eco))/sum(conf_mat_eco),2)
metrics[4,5]  <- round(sensitivity(pred_test, ecoli_test$outbreak),2)
metrics[5,5] <- round(specificity(pred_test, ecoli_test$outbreak),2)
```

Summary of metrics for all models

```
## Testing data ##

# Get the metrics table in a nice format
names(metrics) <- c("", "Risk score; Salmonella", "Tree; Salmonella",
                    "Risk score; E. Coli", "Tree; E. coli")

# Print the table for the report
metrics %>% kbl(caption = "Metrics for models on testing data",
                     booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

Table 7: Metrics for models on testing data

	Risk score; Salmonella	Tree; Salmonella	Risk score; E. Coli	Tree; E. coli
AUC	0.56	0.54	0.79	0.58
Brier Score	0.18	0.25	0.08	0.19
Accuracy	0.76	0.70	0.91	0.72
Sensitivity	0.99	0.91	1.00	0.76
Specificity	0.00	0.00	0.00	0.27

```
## Training data ##
names(metrics_train) <- c("", "Risk score; Salmonella", "Tree; Salmonella",
                    "Risk score; E. Coli", "Tree; E. coli")

# Print the table for the report
metrics_train %>% kbl(caption = "Metrics for models on training data",
                     booktabs=T, escape=T, align = "c") %>%
                     kable_styling(full_width = FALSE, latex_options = c('hold_position'))
```

Appendix

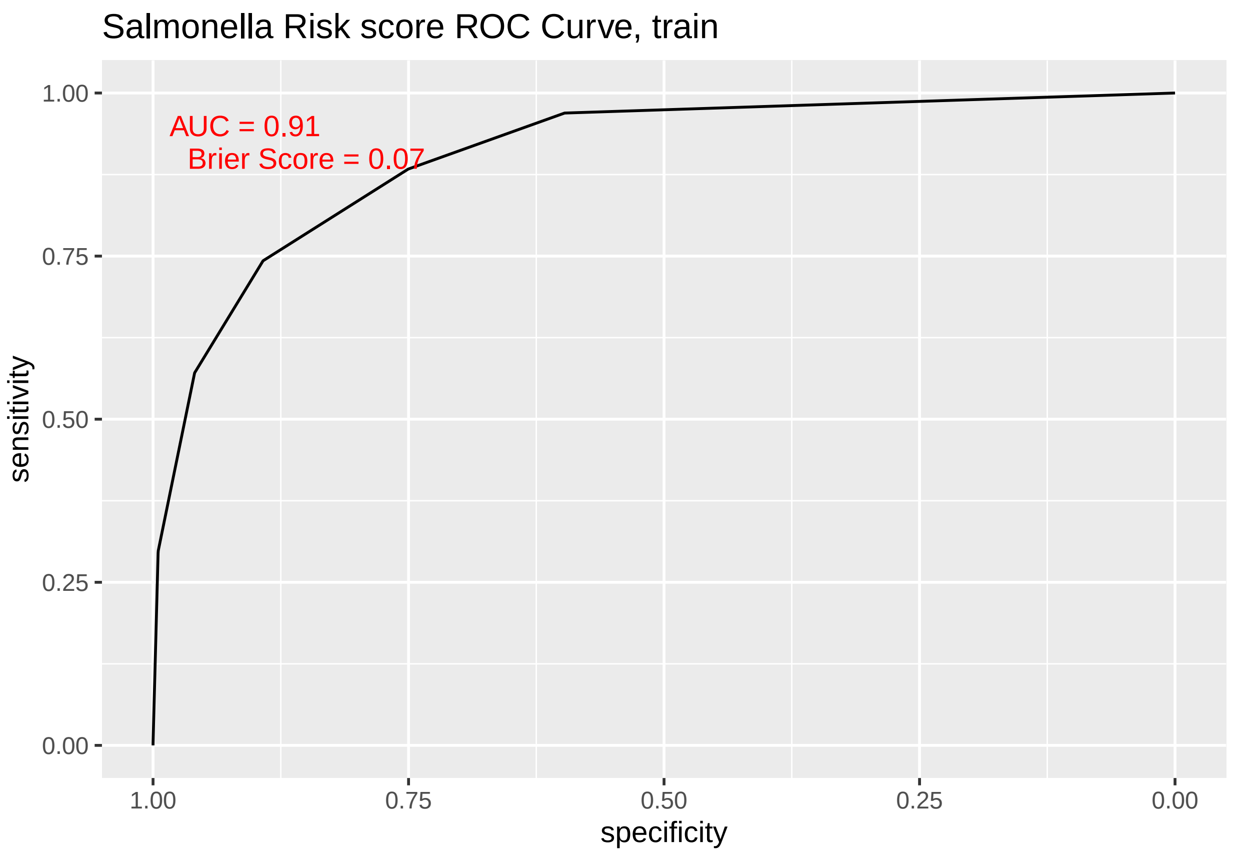
This appendix contains ROC curves for the different models that were fit:

```
#~~~~~~~~~~~~~~#
## Salmonella ##
#~~~~~~~~~~~~~~#
```

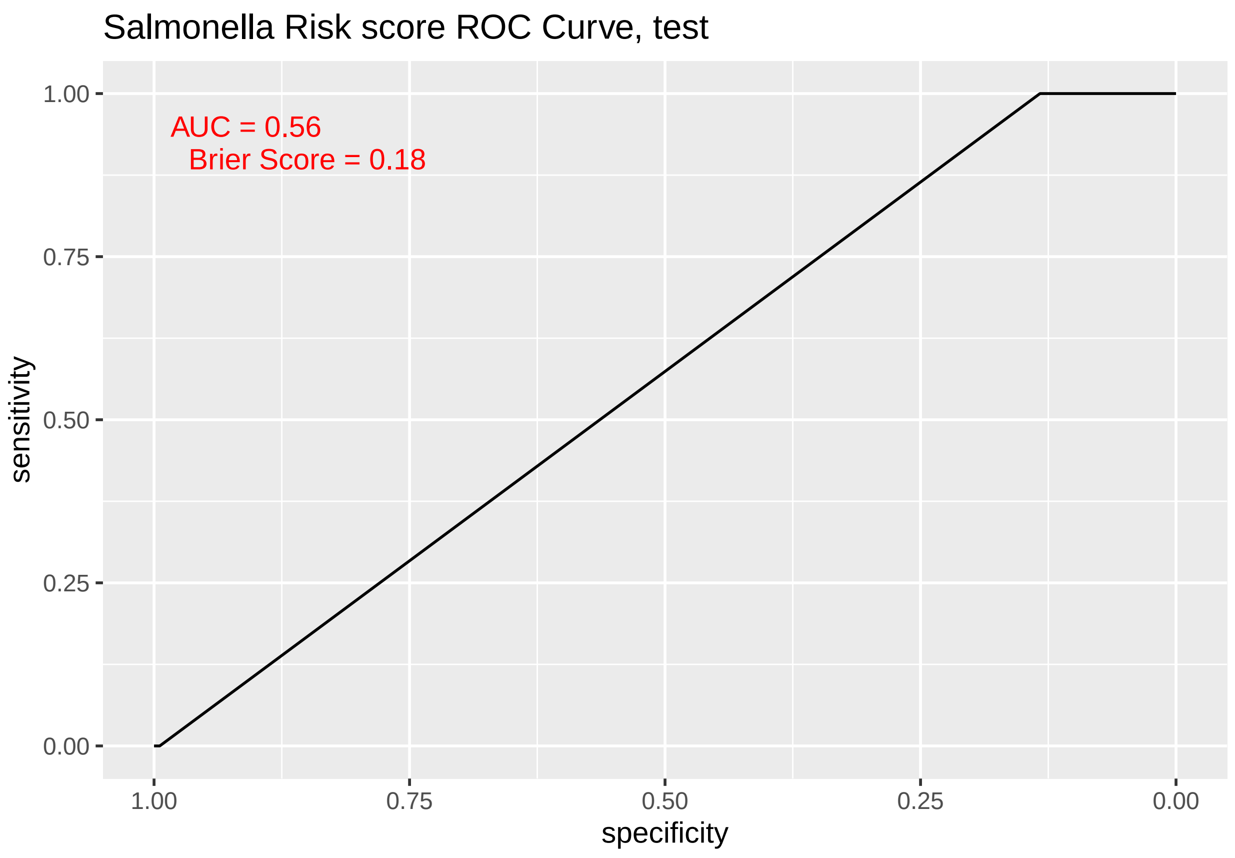
Table 8: Metrics for models on training data

	Risk score; Salmonella	Tree; Salmonella	Risk score; E. Coli	Tree; E. coli
AUC	0.91	0.88	0.95	0.97
Brier Score	0.07	0.13	0.06	0.06
Accuracy	0.90	0.86	0.93	0.92
Sensitivity	0.96	0.86	0.96	0.91
Specificity	0.57	0.83	0.77	0.96

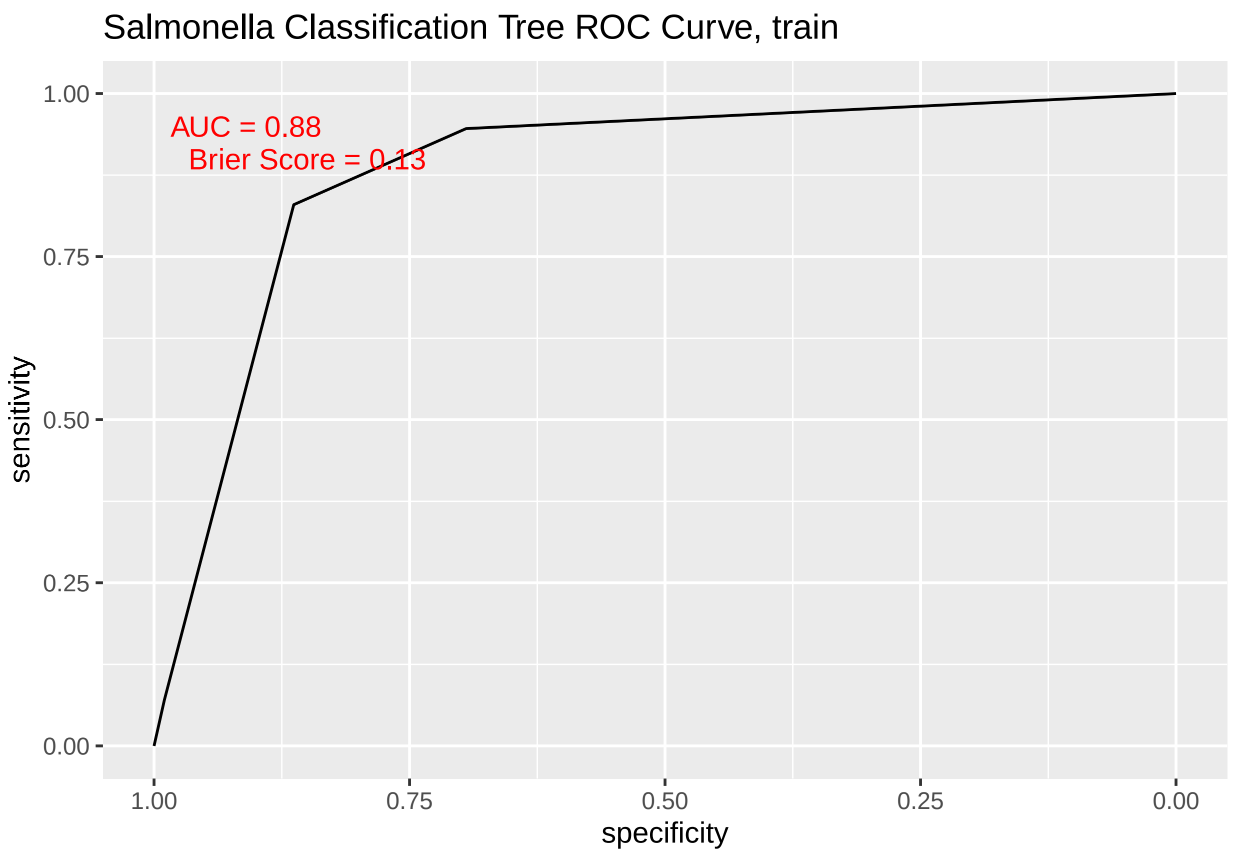
```
# Salmonella Training ROC, Risk score model
ggroc(risk_perfomance_salmonella_train) + ggtitle('Salmonella Risk score ROC Curve, train') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', round(auc(risk_perfomance_salmonella_trai
         color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', round(BrierScore(mod_sal_train),2)
         , color = 'red')
```



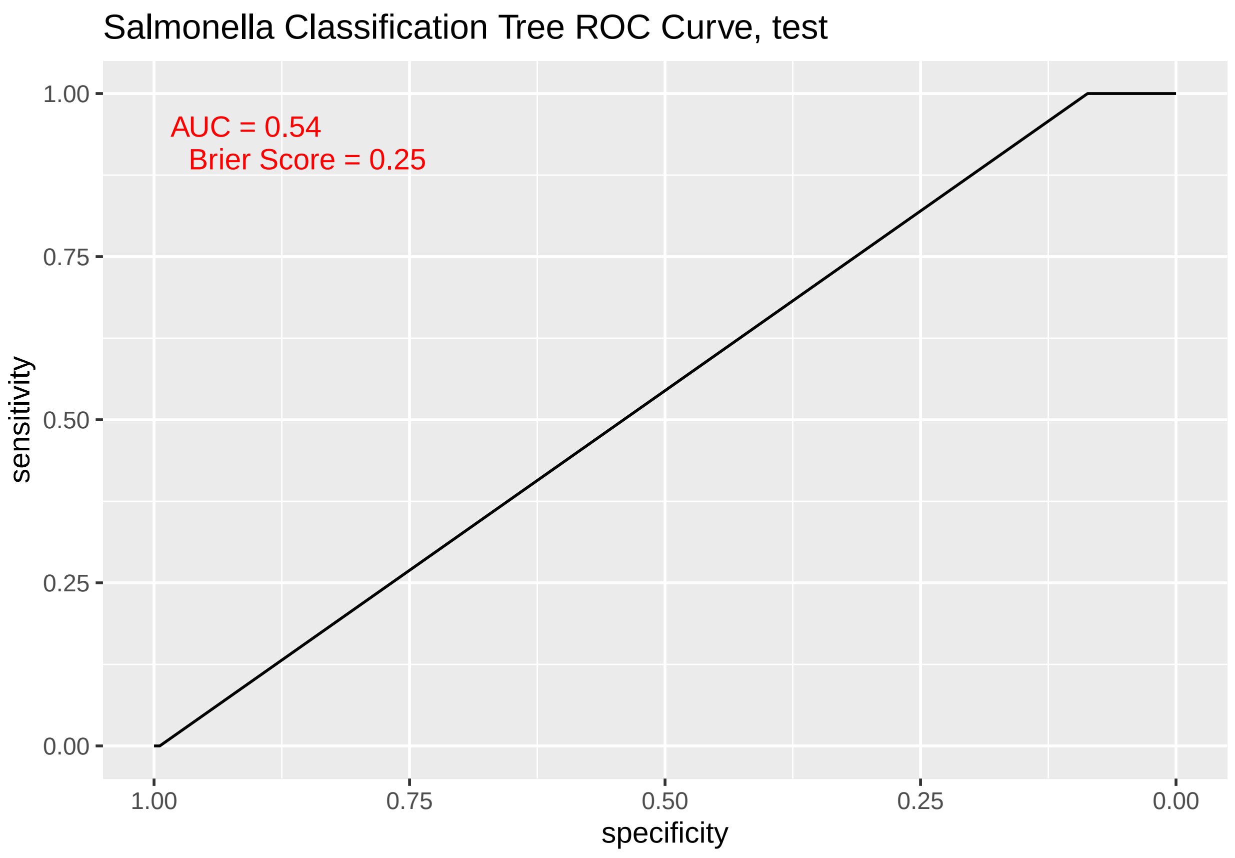
```
# Salmonella Testing ROC, Risk score model
ggroc(risk_perfomance_salmonella) + ggtitle('Salmonella Risk score ROC Curve, test') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', round(auc(risk_perfomance_salmonella),2))
         color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', round(BrierScore(mod_salmonella),2
         , color = 'red')
```



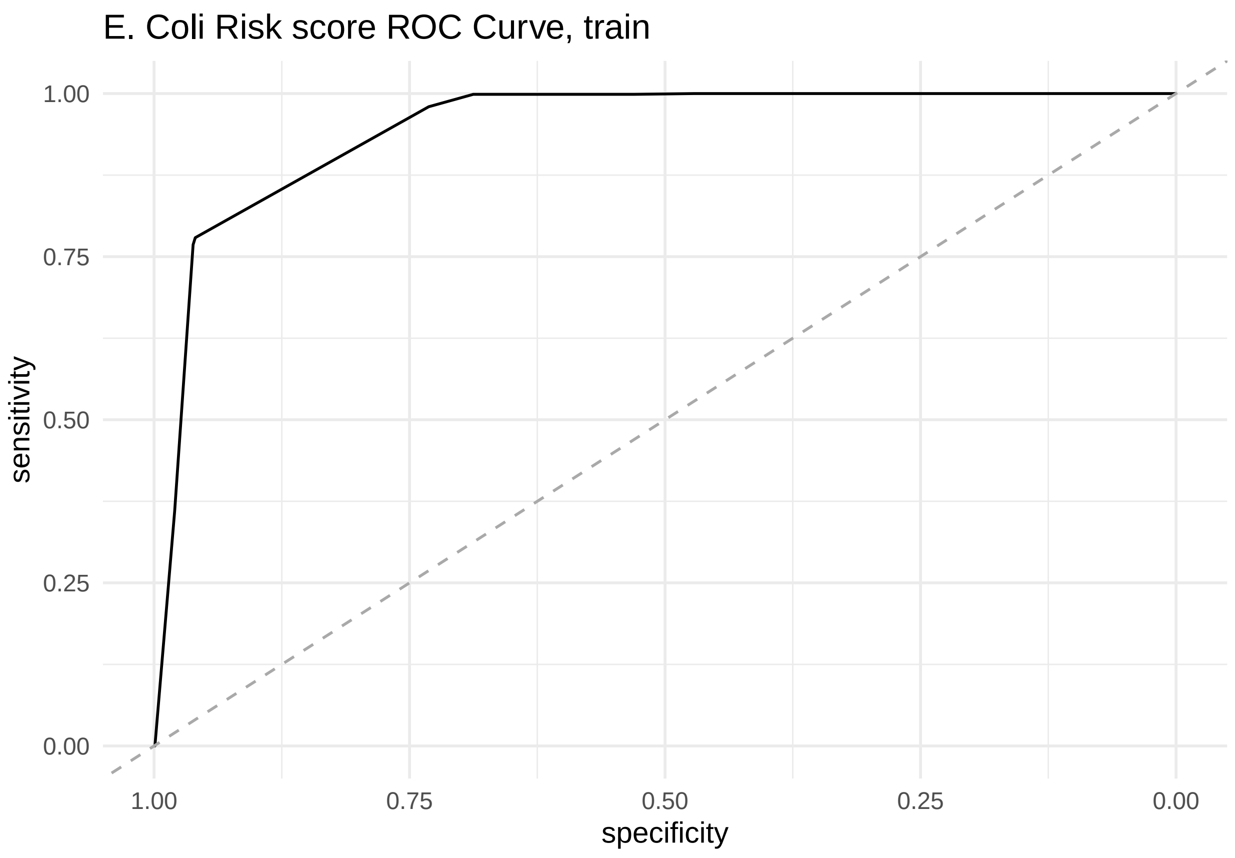
```
# Salmonella Training ROC, tree based model
ggroc(salmonella_acc) + ggtitle('Salmonella Classification Tree ROC Curve, train') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', salmonella_AUC), color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', salmonella_Brscr), color = 'red')
```



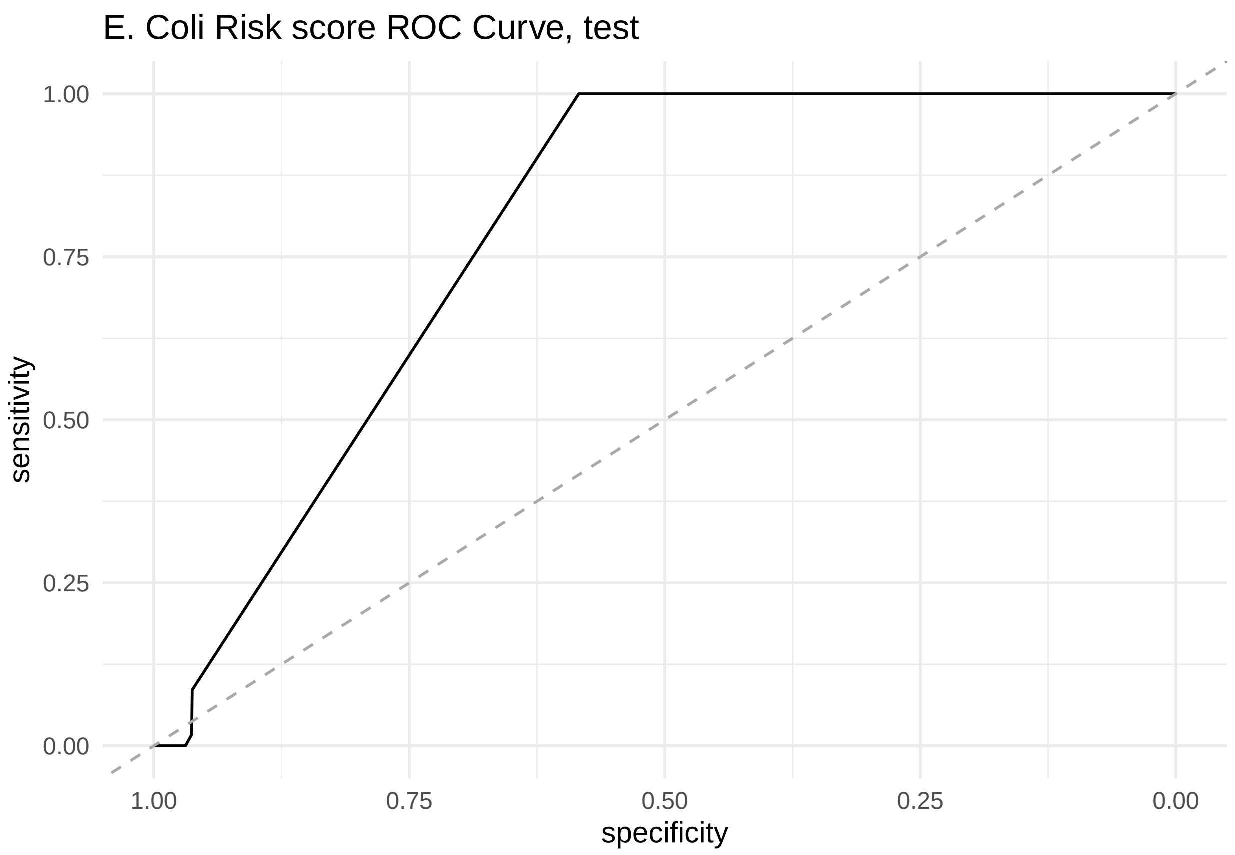
```
# Salmonella Testing ROC, tree based model
ggroc(salmonella_test_acc) + ggtitle('Salmonella Classification Tree ROC Curve, test') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', salmonella_test_AUC), color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', salmonella_test_Brscr), color = 'r
```



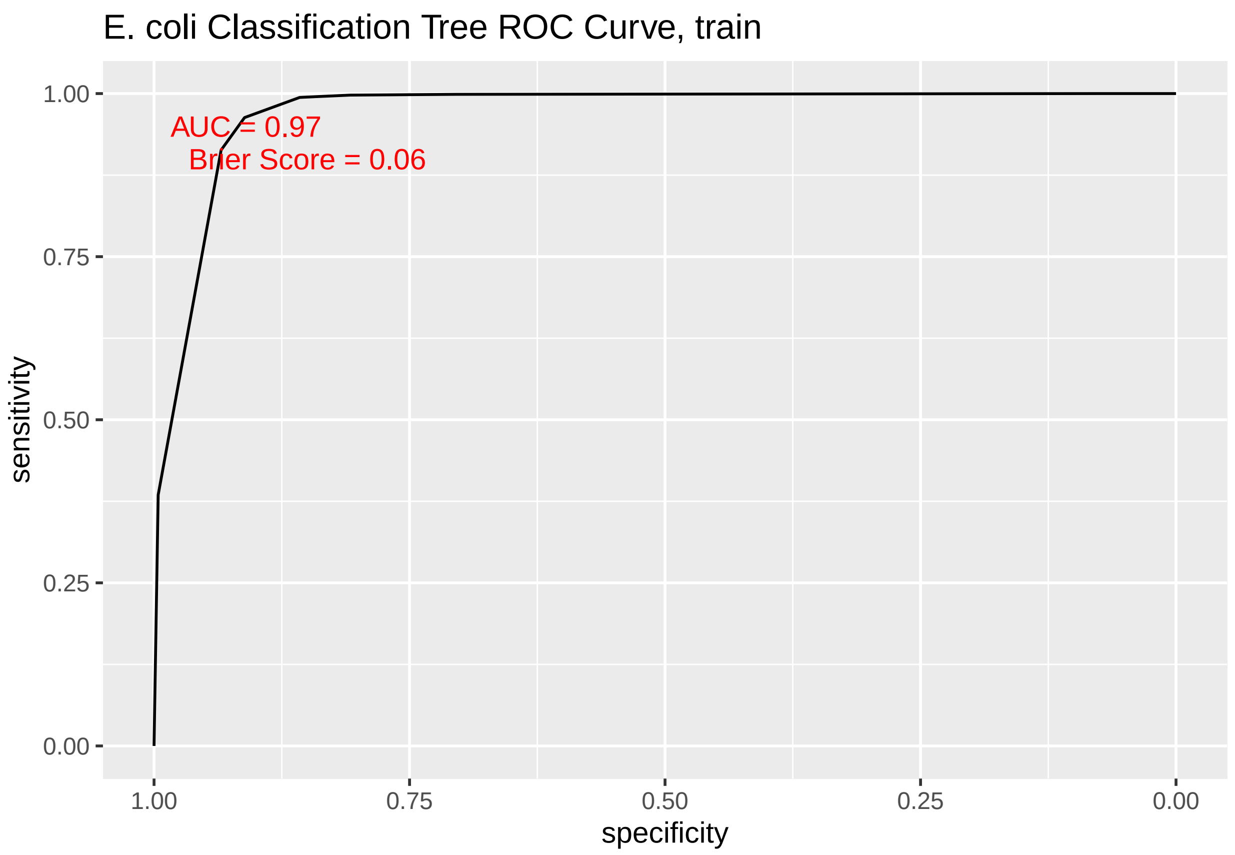
```r
#~~~~~~~~~~~~#
## E. Coli ##
#~~~~~~~~~~~~#

# E. coli, Risk score, training
ggroc(risk_perfomance_ecoli_train) +
  ggtitle('E. Coli Risk score ROC Curve, train') +
  theme_minimal() +
  geom_abline(intercept = 1, slope = 1,
              color = "darkgrey", linetype = "dashed")
```



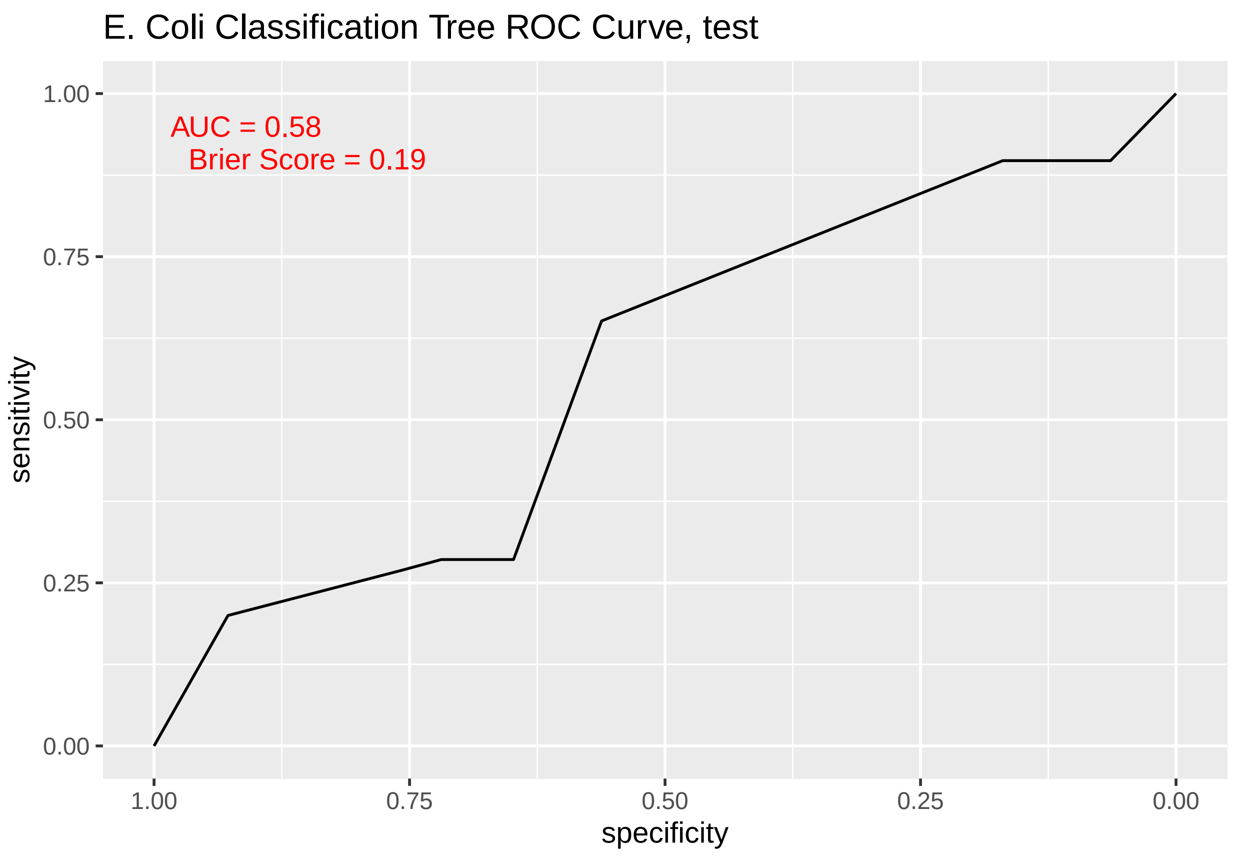
```r
# E. coli, Risk score, testing
ggroc(risk_perfomance_ecoli) +
  ggtitle('E. Coli Risk score ROC Curve, test') +
  theme_minimal() +
  geom_abline(intercept = 1, slope = 1,
              color = "darkgrey", linetype = "dashed")
```



```
# E. Coli Training ROC, tree based model
ggroc(ecoli_acc) + ggtitle('E. coli Classification Tree ROC Curve, train') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', ecoli_AUC), color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', ecoli_Brscr), color = 'red')
```



```
# E. Coli Testing ROC, tree based model
ggroc(ecoli_test_acc) + ggtitle('E. Coli Classification Tree ROC Curve, test') +
  annotate("text", x = .91, y = .95, label = paste0('AUC = ', ecoli_test_AUC), color = 'red') +
  annotate("text", x = .85, y = .9, label = paste0('Brier Score = ', ecoli_test_Brscr), color = 'red')
```


E. Coli Classification Tree ROC Curve, test
AUC = 0.58
Brier Score = 0.19
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00