Thesis

Yi Yang

2023-12-18

```
library(readxl)
library(pander)
library(ggplot2)
library(moments)
library(tidyverse)
library(psych)
library(rio)
library(MASS)
library(ResourceSelection)
library(car)
library(VGAM)
library(pROC)
library(lmtest)
library(fastDummies)
library(table1)
library(haven)
library(caret)
library(corrplot)
library(r02pro)
library(plyr)
library(tree)
library(gbm)
library(leaps)
library(readr)
library(glmnet)
library(dplyr)
```

```
data <- read_excel("Her-2  2+ FISH(-).xlsx");
head(data)
```

```
## # A tibble: 6 x 39
##          ~1   ~2    CA125 CA15-~3  ~4   ~5  ~6
##   <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl>   <dbl>  <dbl>    <dbl>  <dbl>  <dbl>
## 1     3     1      0        1    58  13.1   10.6      61        0      4      2
## 2     3     2      0        1    70  13.5    5.97    123        0      7      2
## 3     3     3      0        1    59   4.1   10.4      86        0      4      2
## 4     3     4      0        1    75  59.9    9.7      63        0      4      2
## 5     3     5      0        1    71  19.0   19.9     102        0      4      2
## 6     3     6      0        1    51  16.6    8.33     48        0      4      2
## # ... with 28 more variables:     <dbl>,     <dbl>,
## #   ` 2cm` <dbl>, T  <dbl>, N  <dbl>, TNM  <dbl>,    <chr>,
```

```
## #       <dbl>, ` (LN)` <dbl>, ER <dbl>, PR <dbl>, HER2 <dbl>,
## #   FISH <chr>, Ki67 <dbl>,    <dbl>,    <dbl>,    <dbl>,
## #     <dbl>,   <dbl>,   <dbl>,   <dbl>,   <dbl>,   <dbl>,   <dbl>,
## #   `AE\r\n ` <dbl>, AE <dbl>, KI67 <dbl>,    <dbl>, and abbreviated
## #   variable names 1:    , 2: `  45`, 3: `CA15-3`, ...
```

```
#data <- data[,5:29]

names(data)[names(data) == " "] <- "metastasis"
names(data)[names(data) == " "] <- "age"
names(data)[names(data) == "   -   "] <- "interval"
names(data)[names(data) == "   "] <- "new_assistance"
names(data)[names(data) == "  "] <- "survey_type"
names(data)[names(data) == "  "] <- "Cancer_type"
names(data)[names(data) == "    "] <- "Neural_invasion"
names(data)[names(data) == "    "] <- "Lymphatic_or_blood_vascular_tumor_emboli"
names(data)[names(data) == "  "] <- "Cancer_type"
names(data)[names(data) == "  2cm"] <- "Size_greater_than_2"
names(data)[names(data) == "T "] <- "T_stage"
names(data)[names(data) == "N "] <- "M_stage"
names(data)[names(data) == "TNM "] <- "TNM_stage"
names(data)[names(data) == "   "] <- "Tumor_size"
names(data)[names(data) == "    "] <- "Number_of_lymph_nodes"
names(data)[names(data) == "   "] <- "Molecular_typing"
names(data)[names(data) == " (LN)"] <- "LN"
names(data)[names(data) == "PR"] <- "PR"
names(data)[names(data) == "ER"] <- "ER"
data <- data %>% mutate(Tumor_size = as.numeric(Tumor_size))
data <- data %>% mutate(interval = as.numeric(interval))
head(data)
```

```
## # A tibble: 6 x 39
##             `  45`   age CA125 `CA15-3` interval new_as~1 surve~2
##   <dbl> <dbl>    <dbl>      <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>   <dbl>
## 1     3     1        0          1    58  13.1    10.6        61        0       4
## 2     3     2        0          1    70  13.5     5.97      123        0       7
## 3     3     3        0          1    59   4.1    10.4        86        0       4
## 4     3     4        0          1    75  59.9     9.7        63        0       4
## 5     3     5        0          1    71  19.0    19.9       102        0       4
## 6     3     6        0          1    51  16.6     8.33       48        0       4
## # ... with 29 more variables: Cancer_type <dbl>, Neural_invasion <dbl>,
## #   Lymphatic_or_blood_vascular_tumor_emboli <dbl>, Size_greater_than_2 <dbl>,
## #   T_stage <dbl>, M_stage <dbl>, TNM_stage <dbl>, Tumor_size <dbl>,
## #   Number_of_lymph_nodes <dbl>, LN <dbl>, ER <dbl>, PR <dbl>, HER2 <dbl>,
## #   FISH <chr>, Ki67 <dbl>, Molecular_typing <dbl>,    <dbl>,
## #     <dbl>, metastasis <dbl>,   <dbl>,   <dbl>,   <dbl>,   <dbl>,
## #    <dbl>,   <dbl>, `AE\r\n ` <dbl>, AE <dbl>, KI67 <dbl>, ...
```

```
data <- data[,c(5:22, 25:26, 29, 38)]
head(data)
```

```
## # A tibble: 6 x 22
##     age CA125 `CA15-3` interval new_as~1 surve~2 Cance~3 Neura~4 Lymph~5 Size_~6
```

```
##   <dbl> <dbl>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1    58  13.1    10.6        61        0       4       2       0       0       0
## 2    70  13.5     5.97      123        0       7       2       0       1       0
## 3    59   4.1    10.4        86        0       4       2       0       0       0
## 4    75  59.9     9.7        63        0       4       2       0       0       0
## 5    71  19.0    19.9       102        0       4       2       0       1       1
## 6    51  16.6     8.33       48        0       4       2       0       0       1
## # ... with 12 more variables: T_stage <dbl>, M_stage <dbl>, TNM_stage <dbl>,
## #   Tumor_size <dbl>, Number_of_lymph_nodes <dbl>, LN <dbl>, ER <dbl>,
## #   PR <dbl>, Ki67 <dbl>, Molecular_typing <dbl>, metastasis <dbl>, KI67 <dbl>,
## #   and abbreviated variable names 1: new_assistance, 2: survey_type,
## #   3: Cancer_type, 4: Neural_invasion,
## #   5: Lymphatic_or_blood_vascular_tumor_emboli, 6: Size_greater_than_2
```

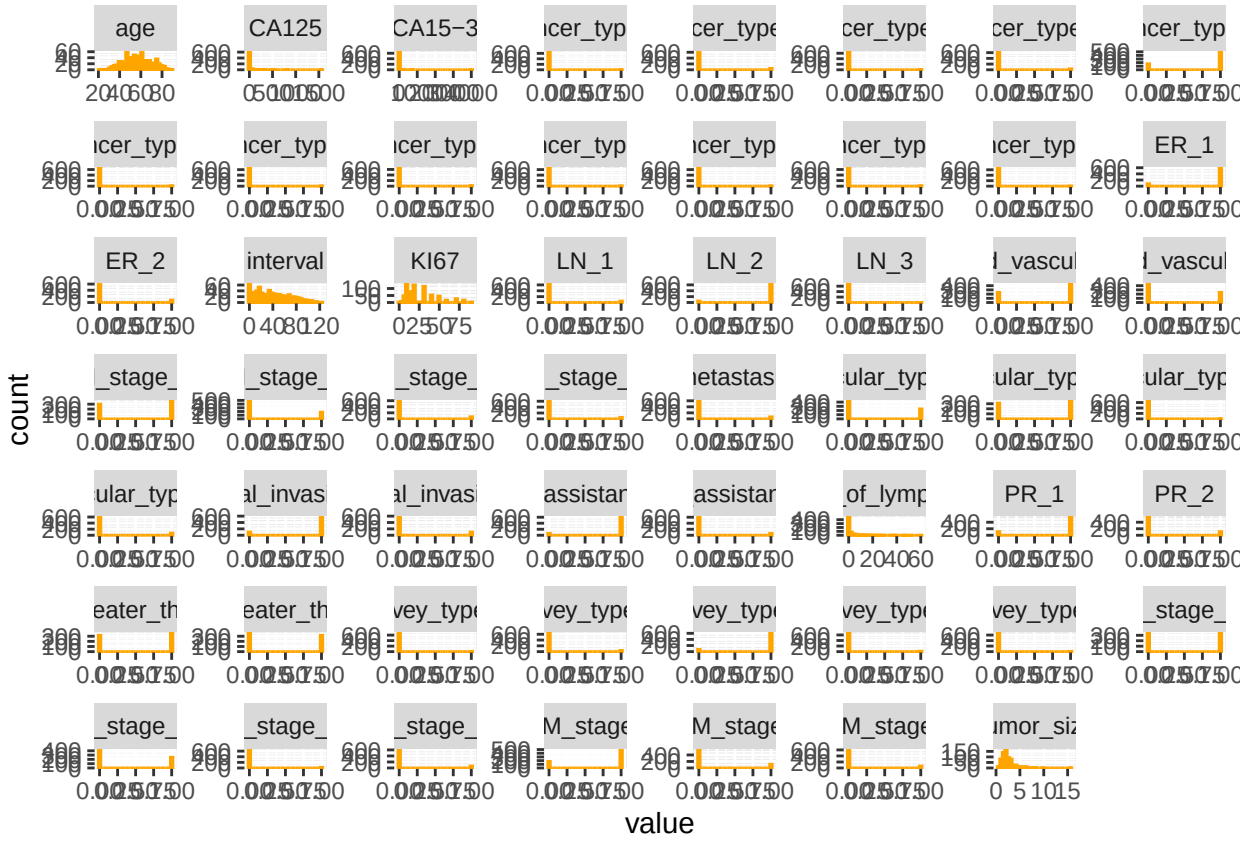
```
data_2 <- dummy_cols(data,select_columns =
                           c( 'new_assistance', 'survey_type', 'Cancer_type',
                             'Neural_invasion','Lymphatic_or_blood_vascular_tumor_emboli',
                             'Size_greater_than_2', 'T_stage', 'M_stage', 'TNM_stage',
                             'Molecular_typing', 'LN', 'PR', 'ER'))
data_3 <- data_2 %>% dplyr::select(-new_assistance, -survey_type,
                                          -Cancer_type, -Neural_invasion,
                                          -Lymphatic_or_blood_vascular_tumor_emboli,
                                          -Size_greater_than_2, -T_stage,
                                          -M_stage, -TNM_stage, -Molecular_typing, -LN,
                                          -PR, -ER,-Ki67)
data_3$metastasis[is.na(data_3$metastasis)] <- 0;
data_3$interval[is.na(data_3$interval)] <- 0;
#data <- data[,c(1:20, 22)]
dim(data_3)
```

```
## [1] 676  55
```

```
table(data_3$metastasis)
```

```
##
##   0   1
## 609  67
```

```
data_3 %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram(col = 'orange')
```




```
dim(data_3)
```

```
## [1] 676  55
```

```
table1(~.|metastasis , data = data_3)
```

```
## Warning in table1.formula(~. | metastasis, data = data_3): Terms to the right
## of '|' in formula 'x' define table columns and are expected to be factors with
## meaningful labels.
```

	0	1	Overall
	(N=609)	(N=67)	(N=676)
age			
Mean (SD)	57.1 (12.7)	56.4 (14.1)	57.0 (12.9)
Median [Min, Max]	57.0 [22.0, 90.0]	57.0 [30.0, 84.0]	57.0 [22.0, 90.0]
CA125			
Mean (SD)	12.3 (12.3)	86.2 (228)	19.6 (75.5)
Median [Min, Max]	9.92 [2.30, 196]	15.4 [3.60, 1560]	10.1 [2.30, 1560]
CA15-3			
Mean (SD)	10.1 (8.03)	97.6 (492)	18.8 (156)
Median [Min, Max]	8.30 [0.300, 101]	13.4 [2.00, 4000]	8.48 [0.300, 4000]
interval			
Mean (SD)	38.9 (29.0)	40.3 (28.9)	39.0 (29.0)

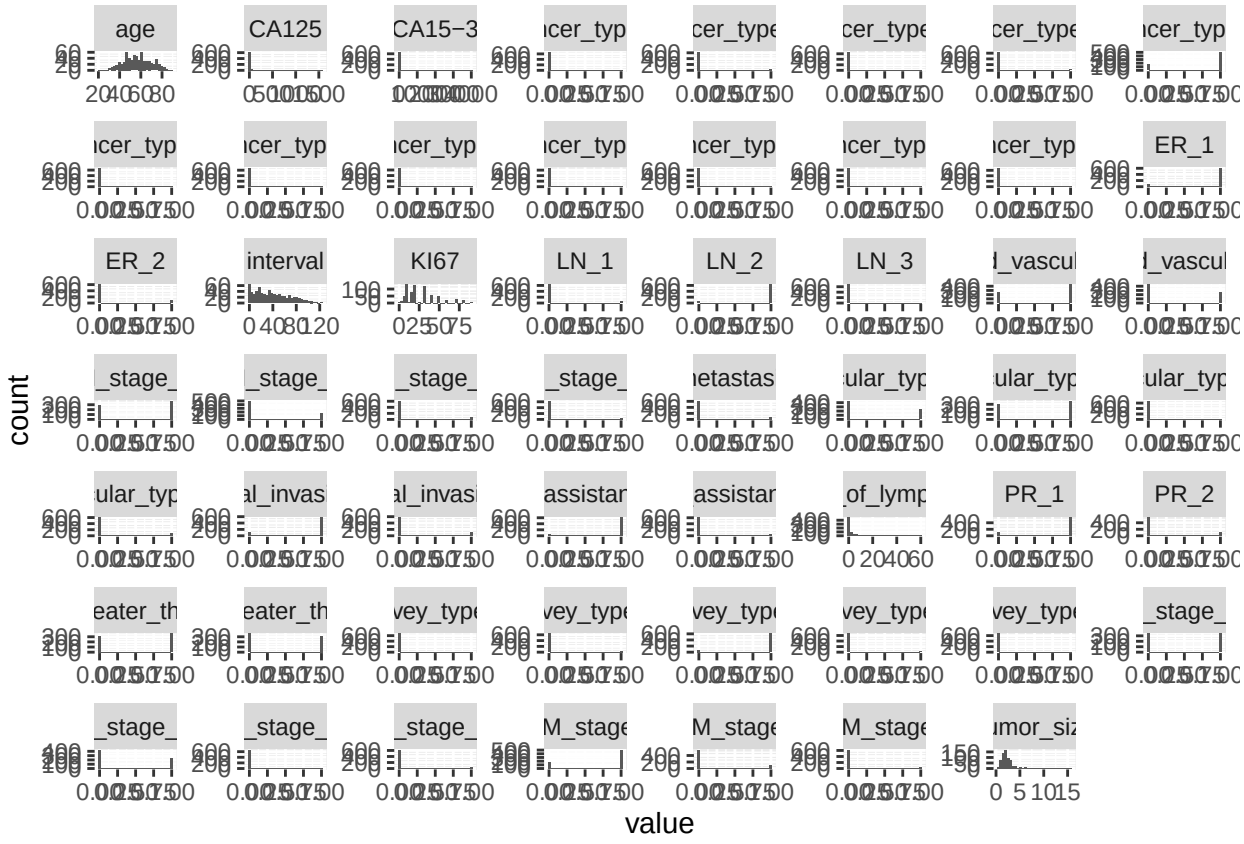
	0	1	Overall
Median [Min, Max]	35.0 [0, 123]	40.0 [0, 113]	35.0 [0, 123]
Tumor_size			
Mean (SD)	2.34 (1.23)	3.05 (2.06)	2.41 (1.35)
Median [Min, Max]	2.00 [0.400, 10.0]	2.50 [0.800, 15.5]	2.00 [0.400, 15.5]
Number_of_lymph_nodes			
Mean (SD)	2.29 (5.99)	4.87 (7.47)	2.54 (6.19)
Median [Min, Max]	0 [0, 60.0]	2.00 [0, 38.0]	0 [0, 60.0]
KI67			
Mean (SD)	25.7 (19.0)	31.5 (20.1)	26.3 (19.2)
Median [Min, Max]	20.0 [0, 90.0]	30.0 [3.00, 90.0]	20.0 [0, 90.0]
new_assistance_0			
Mean (SD)	0.911 (0.285)	0.836 (0.373)	0.904 (0.295)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
new_assistance_1			
Mean (SD)	0.0887 (0.285)	0.164 (0.373)	0.0962 (0.295)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
survey_type_1			
Mean (SD)	0.00985 (0.0988)	0.0299 (0.171)	0.0118 (0.108)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
survey_type_3			
Mean (SD)	0.0624 (0.242)	0.0448 (0.208)	0.0607 (0.239)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
survey_type_4			
Mean (SD)	0.888 (0.315)	0.910 (0.288)	0.891 (0.312)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
survey_type_5			
Mean (SD)	0.0230 (0.150)	0.0149 (0.122)	0.0222 (0.147)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
survey_type_7			
Mean (SD)	0.0164 (0.127)	0 (0)	0.0148 (0.121)
Median [Min, Max]	0 [0, 1.00]	0 [0, 0]	0 [0, 1.00]
Cancer_type_1			
Mean (SD)	0.00328 (0.0573)	0.0448 (0.208)	0.00740 (0.0857)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_2			
Mean (SD)	0.750 (0.433)	0.791 (0.410)	0.754 (0.431)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
Cancer_type_3			
Mean (SD)	0.0181 (0.133)	0.0149 (0.122)	0.0178 (0.132)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_4			
Mean (SD)	0.0131 (0.114)	0 (0)	0.0118 (0.108)
Median [Min, Max]	0 [0, 1.00]	0 [0, 0]	0 [0, 1.00]
Cancer_type_5			
Mean (SD)	0.0246 (0.155)	0.0149 (0.122)	0.0237 (0.152)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_6			
Mean (SD)	0.0164 (0.127)	0 (0)	0.0148 (0.121)
Median [Min, Max]	0 [0, 1.00]	0 [0, 0]	0 [0, 1.00]
Cancer_type_7			
Mean (SD)	0.0131 (0.114)	0.0149 (0.122)	0.0133 (0.115)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]

	0	1	Overall
Cancer_type_8			
Mean (SD)	0.0197 (0.139)	0.0149 (0.122)	0.0192 (0.137)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_9			
Mean (SD)	0.00328 (0.0573)	0.0149 (0.122)	0.00444 (0.0665)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_10			
Mean (SD)	0.0722 (0.259)	0.0448 (0.208)	0.0695 (0.255)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_11			
Mean (SD)	0.0148 (0.121)	0.0149 (0.122)	0.0148 (0.121)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Cancer_type_12			
Mean (SD)	0.0509 (0.220)	0.0299 (0.171)	0.0488 (0.216)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Neural_invasion_0			
Mean (SD)	0.860 (0.347)	0.776 (0.420)	0.852 (0.355)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
Neural_invasion_1			
Mean (SD)	0.140 (0.347)	0.224 (0.420)	0.148 (0.355)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Lymphatic_or_blood_vascular_tumor_emboli_0			
Mean (SD)	0.658 (0.475)	0.522 (0.503)	0.645 (0.479)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
Lymphatic_or_blood_vascular_tumor_emboli_1			
Mean (SD)	0.342 (0.475)	0.478 (0.503)	0.355 (0.479)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Size_greater_than_2_0			
Mean (SD)	0.547 (0.498)	0.373 (0.487)	0.530 (0.499)
Median [Min, Max]	1.00 [0, 1.00]	0 [0, 1.00]	1.00 [0, 1.00]
Size_greater_than_2_1			
Mean (SD)	0.453 (0.498)	0.627 (0.487)	0.470 (0.499)
Median [Min, Max]	0 [0, 1.00]	1.00 [0, 1.00]	0 [0, 1.00]
T_stage_1			
Mean (SD)	0.550 (0.498)	0.0448 (0.208)	0.500 (0.500)
Median [Min, Max]	1.00 [0, 1.00]	0 [0, 1.00]	0.500 [0, 1.00]
T_stage_2			
Mean (SD)	0.402 (0.491)	0.0597 (0.239)	0.368 (0.483)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
T_stage_3			
Mean (SD)	0.0328 (0.178)	0.0448 (0.208)	0.0340 (0.181)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
T_stage_4			
Mean (SD)	0.0148 (0.121)	0.851 (0.359)	0.0976 (0.297)
Median [Min, Max]	0 [0, 1.00]	1.00 [0, 1.00]	0 [0, 1.00]
M_stage_0			
Mean (SD)	0.575 (0.495)	0.373 (0.487)	0.555 (0.497)
Median [Min, Max]	1.00 [0, 1.00]	0 [0, 1.00]	1.00 [0, 1.00]
M_stage_1			
Mean (SD)	0.268 (0.443)	0.254 (0.438)	0.266 (0.442)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
M_stage_2			

	0	1	Overall
Mean (SD)	0.0952 (0.294)	0.179 (0.386)	0.104 (0.305)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
M_stage_3			
Mean (SD)	0.0624 (0.242)	0.194 (0.398)	0.0754 (0.264)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
TNM_stage_2			
Mean (SD)	0.813 (0.390)	0.0448 (0.208)	0.737 (0.441)
Median [Min, Max]	1.00 [0, 1.00]	0 [0, 1.00]	1.00 [0, 1.00]
TNM_stage_3			
Mean (SD)	0.172 (0.378)	0.104 (0.308)	0.166 (0.372)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
TNM_stage_4			
Mean (SD)	0.0148 (0.121)	0.851 (0.359)	0.0976 (0.297)
Median [Min, Max]	0 [0, 1.00]	1.00 [0, 1.00]	0 [0, 1.00]
Molecular_typing_1			
Mean (SD)	0.366 (0.482)	0.224 (0.420)	0.352 (0.478)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
Molecular_typing_2			
Mean (SD)	0.527 (0.500)	0.642 (0.483)	0.538 (0.499)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
Molecular_typing_3			
Mean (SD)	0.00328 (0.0573)	0 (0)	0.00296 (0.0544)
Median [Min, Max]	0 [0, 1.00]	0 [0, 0]	0 [0, 1.00]
Molecular_typing_4			
Mean (SD)	0.103 (0.305)	0.134 (0.344)	0.107 (0.309)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
LN_1			
Mean (SD)	0.0805 (0.272)	0.0149 (0.122)	0.0740 (0.262)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
LN_2			
Mean (SD)	0.918 (0.275)	0.985 (0.122)	0.925 (0.264)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
LN_3			
Mean (SD)	0.00164 (0.0405)	0 (0)	0.00148 (0.0385)
Median [Min, Max]	0 [0, 1.00]	0 [0, 0]	0 [0, 1.00]
PR_1			
Mean (SD)	0.847 (0.360)	0.716 (0.454)	0.834 (0.372)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
PR_2			
Mean (SD)	0.153 (0.360)	0.284 (0.454)	0.166 (0.372)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]
ER_1			
Mean (SD)	0.888 (0.315)	0.851 (0.359)	0.885 (0.320)
Median [Min, Max]	1.00 [0, 1.00]	1.00 [0, 1.00]	1.00 [0, 1.00]
ER_2			
Mean (SD)	0.112 (0.315)	0.149 (0.359)	0.115 (0.320)
Median [Min, Max]	0 [0, 1.00]	0 [0, 1.00]	0 [0, 1.00]

```
data_3 %>%
  keep(is.numeric) %>%
  gather() %>%
```

```
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram()
```




```
tr_ind <- 1:(nrow(data_3) * 0.7)
data_tall <- data_3[tr_ind, ]
nrow(data_tall)
```

```
## [1] 473
```

```
data_te <- data_3[-tr_ind, ]
nrow(data_te)
```

```
## [1] 203
```

```
tr_ind2 <- 1:(nrow(data_tall) * 0.7)
data_tr <- data_tall[tr_ind2, ]
nrow(data_tr)
```

```
## [1] 331
```

```r
data_va <- data_tall[-tr_ind2, ]
nrow(data_va)
```

```
## [1] 142
```

Variable Selection BIC

```r
set.seed(0)
fit_BIC <- regsubsets(metastasis ~ ., data = data_tr, really.big=T)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 16 linear dependencies found
```

```
## Reordering variables and trying again:
```

```r
summary_BIC <- summary(fit_BIC)
min_BIC <- which.min(summary_BIC$bic)
min_BIC
```

```
## [1] 4
```

```r
coef_BIC = coef(fit_BIC,min_BIC)
coef_BIC
```

```
##   (Intercept)         CA125      interval          LN_2 Cancer_type_9
##   0.130102026   0.001808566  -0.001353647   0.054684355   0.000000000
```

```r
formula1 <- metastasis ~CA125+interval+LN_2 +Cancer_type_9+Number_of_lymph_nodes+Tumor_size;
```

Backward Stepwise Selection with Cp

```r
fit_BACKWARD <- regsubsets(metastasis ~ Number_of_lymph_nodes + .,
                           data = data_tr, method = "backward",
                           nvmax = ncol(data_tr)-1, really.big=T)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 16 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is
## not a multiple of replacement length
```

```r
summary_BACKWARD <- summary(fit_BACKWARD)
mix_BACKWARD <- which.min(summary_BACKWARD$cp)
mix_BACKWARD
```

```
## [1] 13
```

```r
coef_BACKWARD = coef(fit_BACKWARD, mix_BACKWARD)
coef_BACKWARD
```

```
##        (Intercept)              CA125           interval      survey_type_3
##        0.144512033        0.001920129       -0.001268590       -0.181669323
##      survey_type_4      Cancer_type_3      Cancer_type_4 Molecular_typing_1
##        0.056448951       -0.143766678       -0.092195324       -0.074597785
## Molecular_typing_2               LN_1               PR_1               ER_1
##       -0.042543771        0.187451163       -0.184096970        0.210987168
##      Cancer_type_9 Molecular_typing_4
##        0.000000000        0.000000000
```

```r
formula2 <- metastasis ~ CA125+interval+survey_type_3+survey_type_4+Cancer_type_3+Cancer_type_4+Molecula
```

```r
library(pROC)
set.seed(0)
predict_BIC = glm(formula1,data_tr,family = "binomial")
```

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

```r
pred_BIC = round(predict(predict_BIC,data_te,type = "response"))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```r
roc_bic <- roc(data_te$metastasis,pred_BIC,smooth=F)
auc(roc_bic)
```

```
## Area under the curve: 0.6939
```

```r
predict_BACKWARD  = glm(formula2,data_tr,family = "binomial")
```

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

```r
pred_BACKWARD = round(predict(predict_BACKWARD,data_te,type = "response"))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```r
roc_back <- roc(data_te$metastasis,pred_BACKWARD,smooth=F)
auc(roc_back)
```

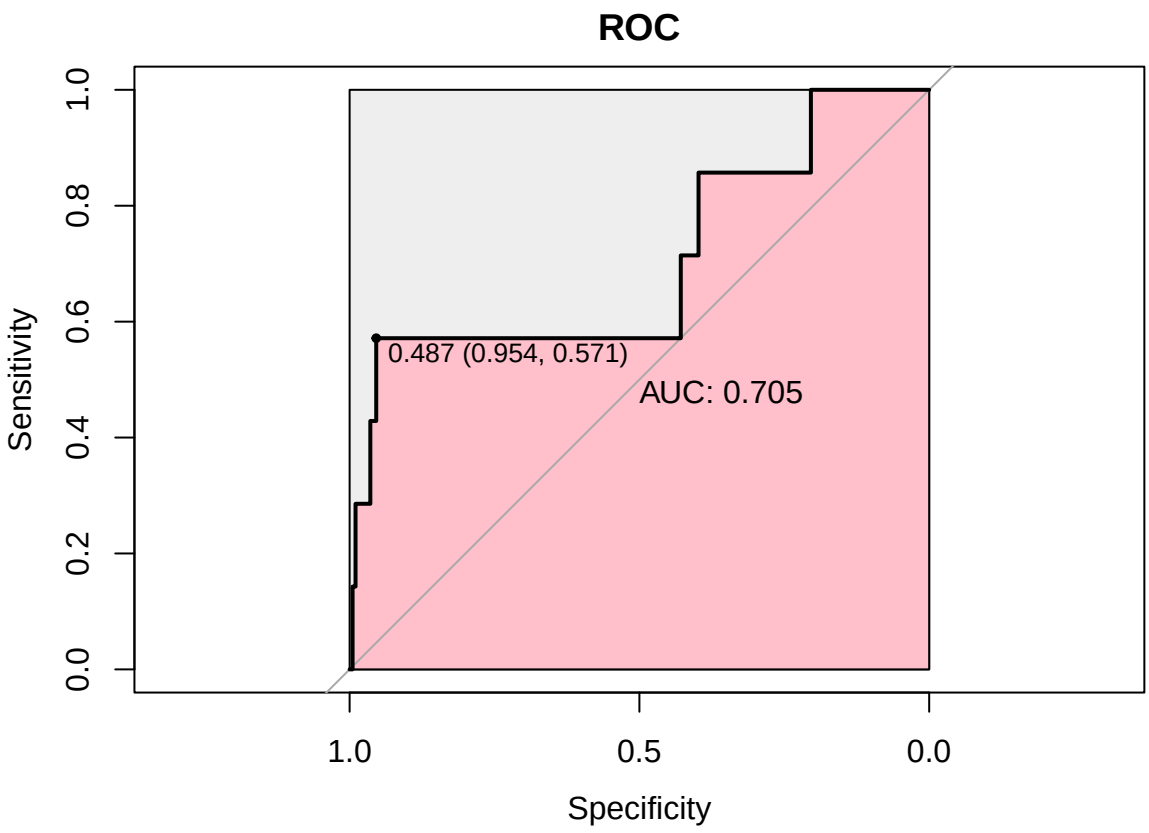
```
## Area under the curve: 0.6148
```

cross validation #Random Forest

```r
library(randomForest)
set.seed(0)
K <- 2
n_all <- nrow(data_tr)
n_all2 <- nrow(data_va)
fold_auc_rf<-as.numeric()
auc_all <- c()
fold_ind <- sample(1:K, n_all, replace = TRUE)
fold_ind2 <- sample(1:K, n_all2, replace = TRUE)

for (i in c(10,100,10)) {
  for (j in 1:K) {
    rf_model <- randomForest(formula1, data = data_tr[fold_ind != j, ], ntree = i,
                             importance = T)
    pred_prob <- predict(rf_model, newdata = data_va[fold_ind2 == j, ], type = "response")
    pred_label <- ifelse(pred_prob > 0.5, 1, 0)
    roc_rf <- roc(data_va[fold_ind2 == j, ]$metastasis,pred_label,smooth=F)
    auc_all[((i/10 -1)*10) + j] <- auc(roc_rf)
  }
}
which.max(auc_all)
```

```
## [1] 2
```

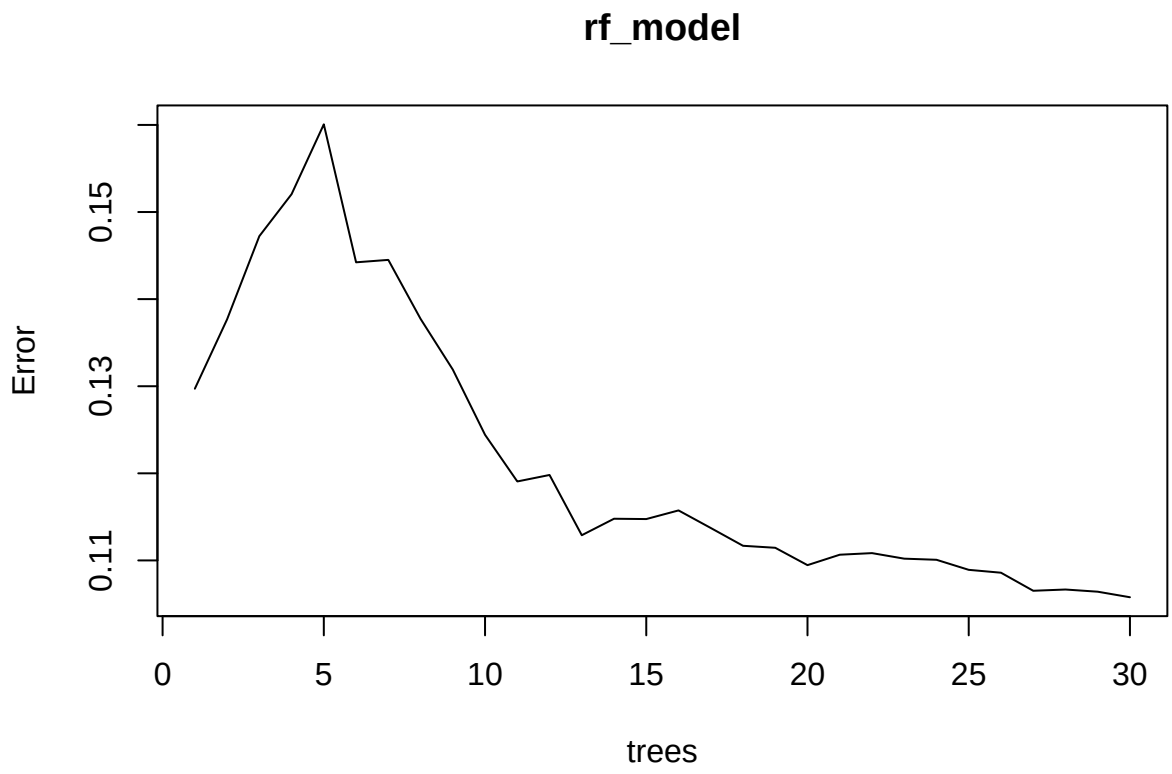
```r
rf_model <- randomForest(formula1, data = data_tr, ntree = 30, importance = T)
pred_prob <- predict(rf_model, newdata = data_te, type = "response")
rf_pred_label <- ifelse(pred_prob > 0.5, 1, 0)
rf_pred<-as.character(pred_prob)
rf_pred<-as.numeric(pred_prob)
rf_roc<-roc(data_te$metastasis,rf_pred,smooth=F)

plot(rf_roc, auc.polygon=T, auc.polygon.col='pink', smooth=F,print.auc=T,
     max.auc.polygon=T,print.thres.cex=0.8, lty=1,main='ROC',print.thres=T)
```




```
rf_auc <- auc(rf_roc)
rf_auc
```

```
## Area under the curve: 0.7048
```

```
plot(rf_model)
```



#KNN

```
set.seed(0)
K <- 2
n_all <- nrow(data_tr)
n_all2 <- nrow(data_va)
fold_auc_rf<-as.numeric()
auc_all3 <- c()
fold_ind <- sample(1:K, n_all, replace = TRUE)
fold_ind2 <- sample(1:K, n_all2, replace = TRUE)

table(data$metastasis)
```

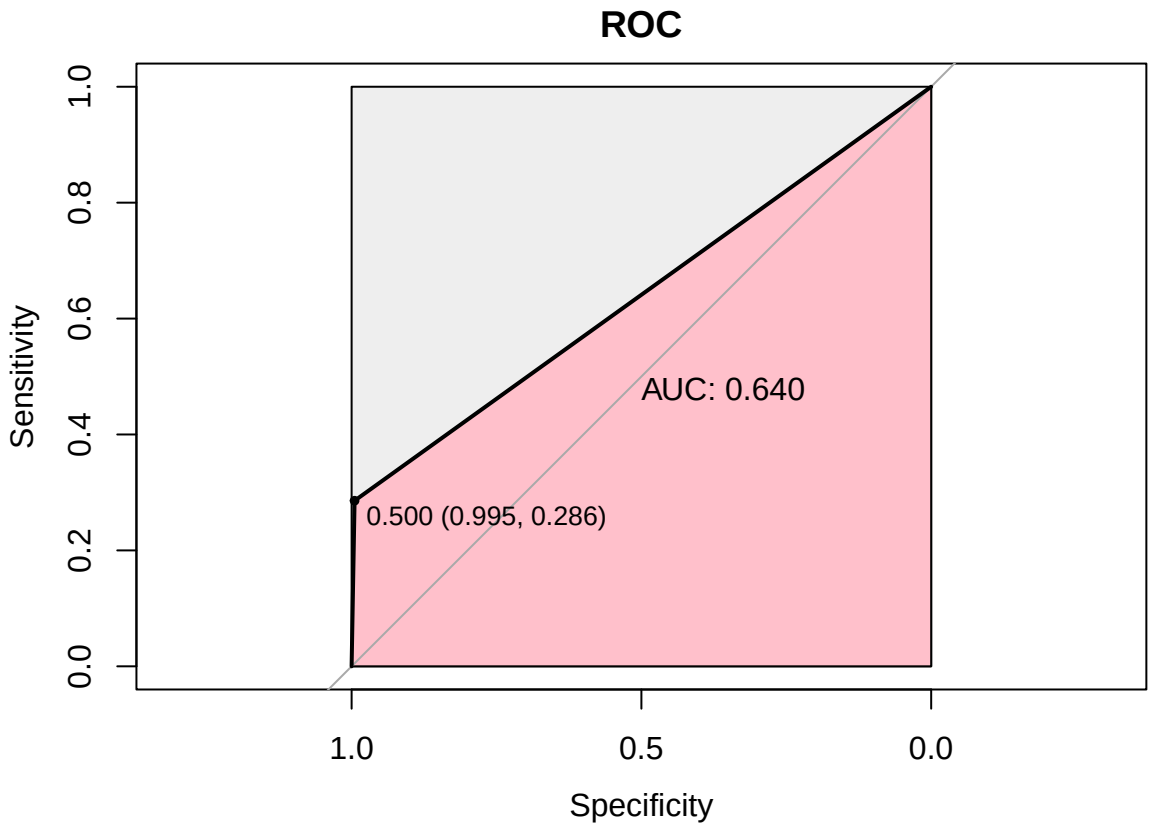
```
##
##  1
## 67
```

```
for (j in 2:K) {
  for (i in 5:20) {
     knn_model <- knn3(formula1, data_tr[fold_ind != j, ], k = i)
    knn_prob <- predict(knn_model, newdata = data_va[fold_ind2 == j, ])
    pred_label <- ifelse(knn_prob > 0.5, 1, 0)
    roc_rf <- roc(data_va[fold_ind2 == j, ]$metastasis,pred_label[,2],smooth=F)
    auc_all3[((j-2)*15) + i - 4] <- auc(roc_rf)
  }
```

```
}
which.max(auc_all3)
```

```
## [1] 2
```

```
knn_model <- knn3(formula1, data_tr, k = 2)
knn_prob <- predict(knn_model, newdata = data_te)
pred_label <- ifelse(knn_prob > 0.5, 1, 0)
knn_roc <- roc(data_te$metastasis,pred_label[,2],smooth=F)
plot(knn_roc, auc.polygon=T, auc.polygon.col='pink', smooth=F,print.auc=T,
     max.auc.polygon=T,print.thres.cex=0.8, lty=1,main='ROC',print.thres=T)
```




```
knn_auc <- auc(knn_roc)
knn_auc
```

```
## Area under the curve: 0.6403
```

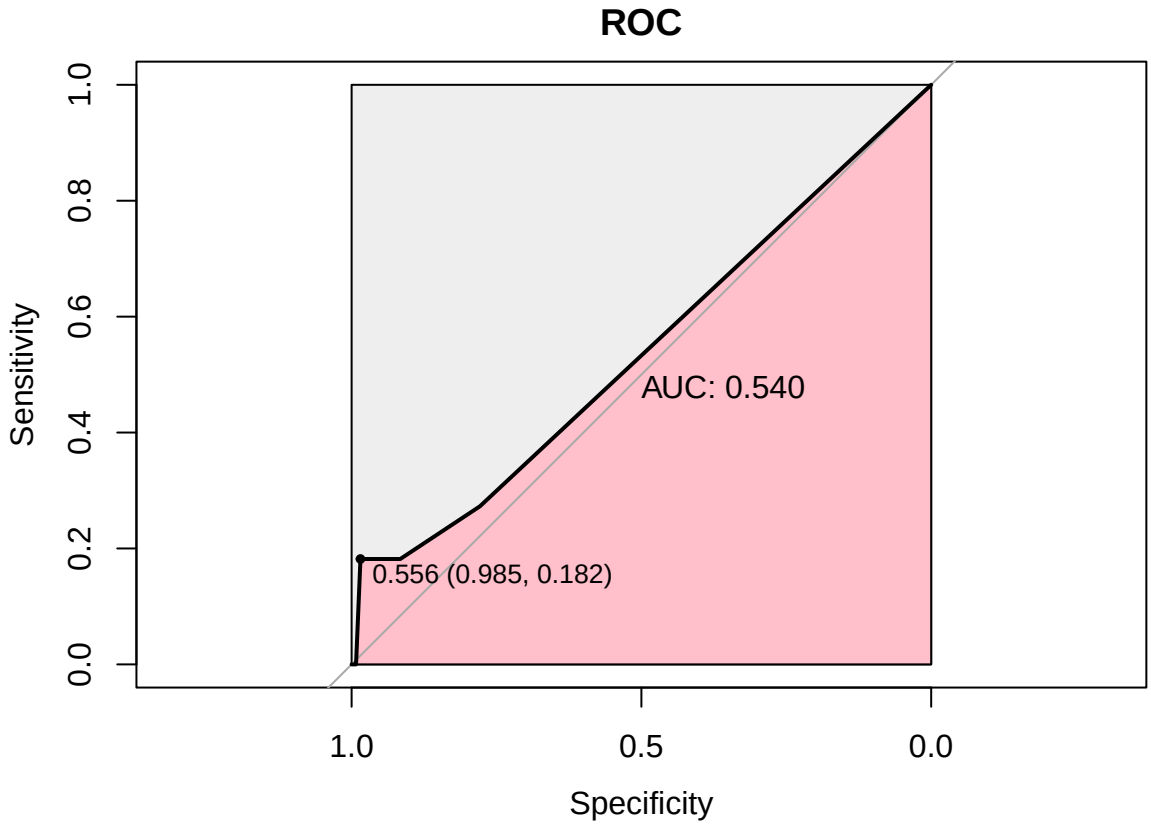
#CART Classification and Regression Trees

```
library(rpart)
set.seed(0)
K <- 2
n_all <- nrow(data_tr)
```

```r
n_all2 <- nrow(data_va)
fold_auc_rf<-as.numeric()
auc_all3 <- c()
fold_ind <- sample(1:K, n_all, replace = TRUE)
fold_ind2 <- sample(1:K, n_all2, replace = TRUE)

for (j in 2:K) {
  for (i in 5:20) {
    rpart_model <- rpart(formula1, data_tr)
    rpart_prob <- predict(rpart_model, newdata = data_va[fold_ind2 == j, ])
    pred_label <- ifelse(rpart_prob > 0.5, 1, 0)
    roc_rf <- roc(data_va[fold_ind2 == j, ]$metastasis, rpart_prob, smooth = FALSE)
    auc_all3[((j - 1) * 15) + i - 4] <- auc(roc_rf)
  }
}
which.max(auc_all3)
```

```
## [1] 16
```

```r
rpart_model <- rpart(formula1, data_tr)
rpart_prob <- predict(rpart_model, newdata = data_va)
pred_label <- ifelse(rpart_prob > 0.5, 1, 0)
roc_rf <- roc(data_va$metastasis, rpart_prob, smooth = FALSE)
plot(roc_rf, auc.polygon=T, auc.polygon.col='pink', smooth=F,print.auc=T,
     max.auc.polygon=T,print.thres.cex=0.8, lty=1,main='ROC',print.thres=T)
```



```
rpart_auc <- auc(roc_rf)
rpart_auc
```

```
## Area under the curve: 0.5399
```

#XGBoost

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.2.3
```

```
set.seed(0)
x_train <- as.matrix(data_tr[, c("CA125","interval","LN_2","Cancer_type_9",
                                 "Number_of_lymph_nodes","Tumor_size")])
x_vali <- as.matrix(data_va[, c("CA125","interval","LN_2","Cancer_type_9",
                                 "Number_of_lymph_nodes","Tumor_size")])
y_train <- data_tr$metastasis
x_test <- as.matrix(data_te[, c("CA125","interval","LN_2","Cancer_type_9",
                                "Number_of_lymph_nodes","Tumor_size")])

auc_all4 <- c()
for (j in 50:100) {
  xgboost_model <- xgboost(data = x_train, label = y_train, nrounds = j,
                           objective = "multi:softmax", num_class = 2, verbose = 0)
  xgboost_prob <- predict(xgboost_model, newdata = as.matrix(x_vali))
  pred_label <- ifelse(xgboost_prob > 0.5, 1, 0)
  roc_xgboost <- roc(data_va$metastasis,pred_label,smooth=F)
  auc_all4[j - 49] <- auc(roc_xgboost)
}
which.min(auc_all4)
```

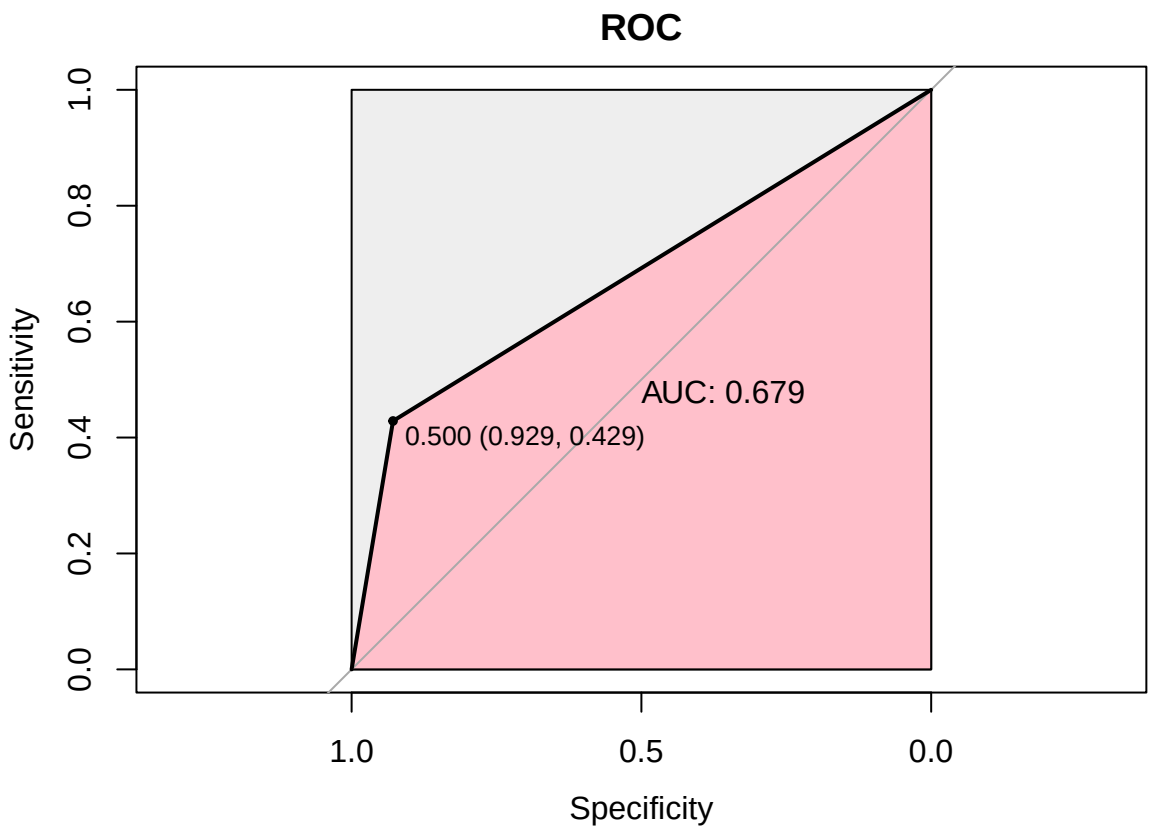
```
## [1] 1
```

```
xgboost_model <- xgboost(data = x_train, label = y_train, nrounds = 50,
                           objective = "multi:softmax", num_class = 2)
```

```
## [1]  train-mlogloss:0.504971
## [2]  train-mlogloss:0.403240
## [3]  train-mlogloss:0.334129
## [4]  train-mlogloss:0.291207
## [5]  train-mlogloss:0.254766
## [6]  train-mlogloss:0.223344
## [7]  train-mlogloss:0.201414
## [8]  train-mlogloss:0.186014
## [9]  train-mlogloss:0.170898
## [10] train-mlogloss:0.162292
## [11] train-mlogloss:0.156985
## [12] train-mlogloss:0.147268
## [13] train-mlogloss:0.138557
## [14] train-mlogloss:0.132827
```

```
## [15] train-mlogloss:0.125353
## [16] train-mlogloss:0.116546
## [17] train-mlogloss:0.113206
## [18] train-mlogloss:0.107830
## [19] train-mlogloss:0.103642
## [20] train-mlogloss:0.093730
## [21] train-mlogloss:0.091468
## [22] train-mlogloss:0.086951
## [23] train-mlogloss:0.082736
## [24] train-mlogloss:0.079518
## [25] train-mlogloss:0.074935
## [26] train-mlogloss:0.073262
## [27] train-mlogloss:0.068602
## [28] train-mlogloss:0.065256
## [29] train-mlogloss:0.063308
## [30] train-mlogloss:0.059759
## [31] train-mlogloss:0.057233
## [32] train-mlogloss:0.054172
## [33] train-mlogloss:0.052589
## [34] train-mlogloss:0.050651
## [35] train-mlogloss:0.049621
## [36] train-mlogloss:0.047557
## [37] train-mlogloss:0.045726
## [38] train-mlogloss:0.043991
## [39] train-mlogloss:0.042780
## [40] train-mlogloss:0.041034
## [41] train-mlogloss:0.039669
## [42] train-mlogloss:0.038628
## [43] train-mlogloss:0.037555
## [44] train-mlogloss:0.036929
## [45] train-mlogloss:0.036216
## [46] train-mlogloss:0.035532
## [47] train-mlogloss:0.034874
## [48] train-mlogloss:0.033918
## [49] train-mlogloss:0.033326
## [50] train-mlogloss:0.032412
```

```r
xgboost_prob <- predict(xgboost_model, newdata = as.matrix(x_test))
pred_label <- ifelse(xgboost_prob > 0.5, 1, 0)
roc_xgboost <- roc(data_te$metastasis,pred_label,smooth=F)
plot(roc_xgboost, auc.polygon=T, auc.polygon.col='pink', smooth=F,print.auc=T,
     max.auc.polygon=T,print.thres.cex=0.8, lty=1,main='ROC',print.thres=T)
```




```
xgboost_auc <- auc(roc_xgboost)
xgboost_auc
```

```
## Area under the curve: 0.6786
```

#Nerual Network

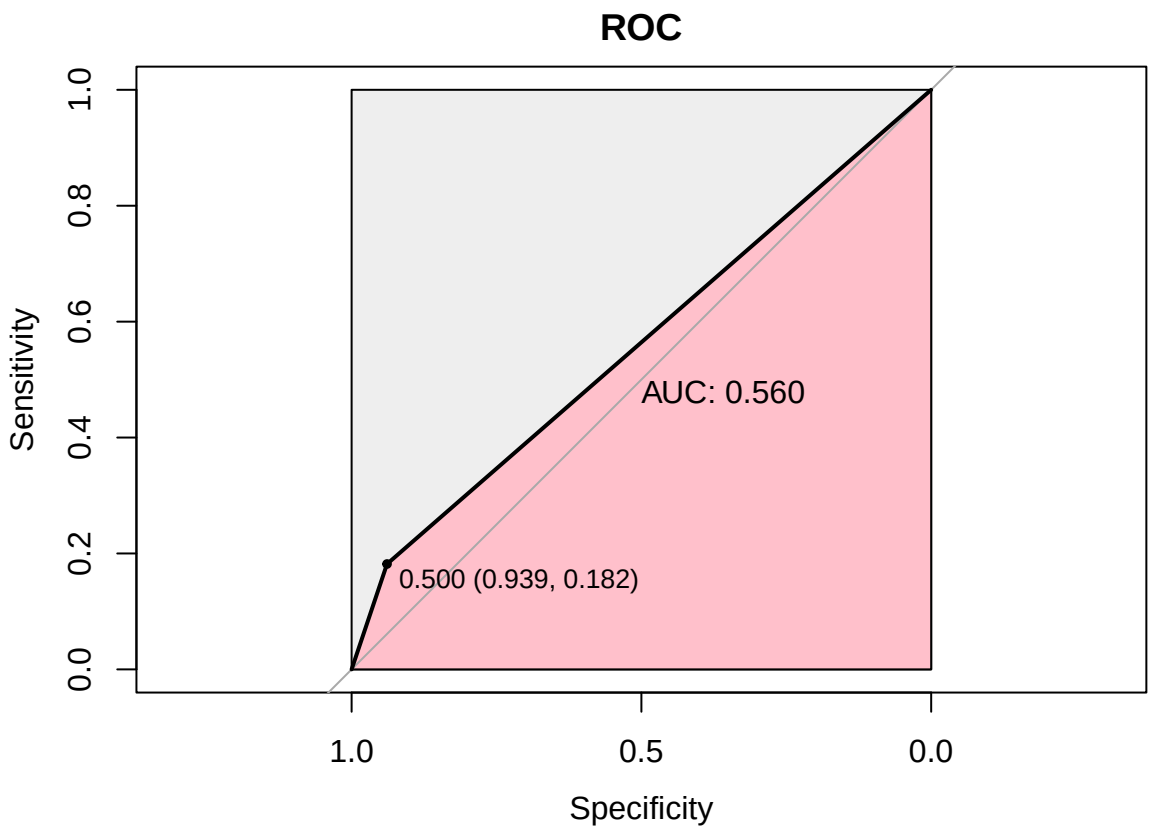
```
set.seed(0)
library(neuralnet)
```

```
## Warning: package 'neuralnet' was built under R version 4.2.3
```

```
nn_model <- neuralnet(formula1, data = data_tr, hidden = c(3, 2), linear.output = FALSE)
nn_prob <- predict(nn_model, data_va)
nn_label <- ifelse(nn_prob > 0.5, 1, 0)
roc_nn <- roc(data_va$metastasis,nn_label,smooth=F)
```

```
## Warning in roc.default(data_va$metastasis, nn_label, smooth = F): Deprecated use
## a matrix as predictor. Unexpected results may be produced, please pass a numeric
## vector.
```

```
plot(roc_nn, auc.polygon=T, auc.polygon.col='pink', smooth=F,print.auc=T,
     max.auc.polygon=T,print.thres.cex=0.8, lty=1,main='ROC',print.thres=T)
```




```
nn_auc <- auc(roc_nn)
nn_auc
```

```
## Area under the curve: 0.5604
```

```
gbm_confusion<-table(data_te$metastasis,rf_pred_label,dnn=c('Actual','Predicted'))
gbm_confusion
```

```
##       Predicted
## Actual   0   1
##      0 188   8
##      1   4   3
```

```
lr_accuracy <- (gbm_confusion[1,1] + gbm_confusion[2,2]) / (gbm_confusion[1,1] + gbm_confusion[1,2] + g

lr_precision <- gbm_confusion[2,2] / (gbm_confusion[2,2] + gbm_confusion[1,2])

lr_recall <- gbm_confusion[2,2] / (gbm_confusion[2,2] + gbm_confusion[2,1])

lr_f1 <- 2/(1/lr_precision + 1/lr_recall)

lr_accuracy
```

```
## [1] 0.9408867
```

```
lr_precision
```

```
## [1] 0.2727273
```

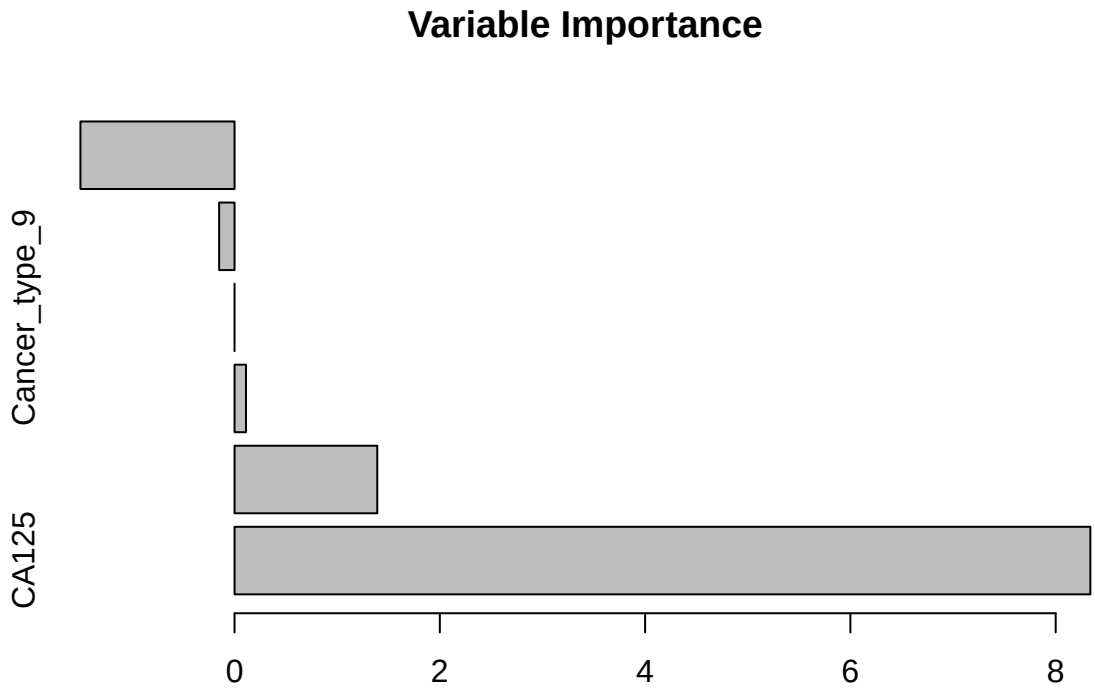
```
lr_recall
```

```
## [1] 0.4285714
```

```
lr_f1
```

```
## [1] 0.3333333
```

```
var_importance <- importance(rf_model)
var_importance <- var_importance[order(var_importance[, 1], decreasing = TRUE), ]
barplot(var_importance[, 1], main = "Variable Importance", horiz = TRUE)
```




```
#CA125+interval+LN_2 +Cancer_type_9+Number_of_lymph_nodes+Tumor_size
var_importance
```

```
##                         %IncMSE IncNodePurity
## CA125                 8.3380258   13.34832773
## Tumor_size            1.3906286    4.92177234
## Number_of_lymph_nodes 0.1117927    3.98269746
## Cancer_type_9         0.0000000    0.00000000
## interval             -0.1502623    6.26359742
## LN_2                 -1.5014895    0.06646368
```

Discussion:

Variable Selection:

Bayesian Information Criterion Bayesian Information Criterion (BIC) is a widely used model selection criterion that balances model fit and complexity by maximizing the likelihood function. It considers the number of parameters in the model and tends to favor simpler models to avoid overfitting. BIC offers several advantages, including its ability to handle model complexity and its consistency under certain conditions, ensuring that the probability of selecting the true model approaches 1 as the sample size approaches infinity. Additionally, BIC is computationally simple, making it easy to understand and implement. However, BIC tends to penalize complex models excessively, potentially overlooking some complex relationships in the data. It is also sensitive to sample size, often selecting overly simple models for small sample sizes, leading to underfitting. Furthermore, BIC's consistency results rely on assumptions such as the true model being among the candidate models and correctly specified, which may not hold in practice.

Backward selection Backward selection is a feature selection method that iteratively fits the model and removes the least significant features to simplify the model. It helps reduce model complexity by eliminating features that have little impact on the response variable, thereby improving the model's interpretability and generalization ability. However, backward selection is a greedy algorithm and may get stuck in local optima, failing to find the global optimal solution. It can also be computationally expensive for large feature sets, as it requires repeatedly fitting the model and evaluating feature significance. Additionally, backward selection typically considers the significance of individual features and may overlook the impact of feature interactions on the model.

Machine Learning

This study utilized random forest, k-nearest neighbors (KNN), classification and regression trees (CART), XGBoost, and neural networks for analysis.

Random Forest Random Forest (RF) is a powerful ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of individual trees. RF offers several advantages, including its ability to handle high-dimensional data and provide estimates of variable importance, which can aid in feature selection. Additionally, RF is less prone to overfitting compared to individual decision trees due to its ensemble nature. However, RF can be computationally expensive for large datasets, and its performance may degrade with highly imbalanced class distributions.

K-Nearest Neighbors K-Nearest Neighbors (KNN) is a simple yet effective non-parametric classification and regression method. KNN makes predictions based on the majority class or average value of the k nearest data points in the feature space. One advantage of KNN is its simplicity and ease of implementation, making it suitable for various applications. However, KNN can be computationally expensive for large datasets, as it requires calculating distances between the target point and all other points in the dataset. Additionally, KNN is sensitive to irrelevant or redundant features and requires careful selection of the distance metric and the value of k.

Classification and Regression Trees Classification and Regression Trees (CART) are a popular decision tree algorithm used for both classification and regression tasks. CART recursively partitions the feature space into regions that minimize impurity, such as Gini impurity or entropy. One key advantage of CART is its interpretability, as the resulting tree can be visualized and easily understood. However, CART is prone to overfitting, especially with complex datasets, and can be unstable, leading to different trees with slight variations in the training data.

XGBoost XGBoost is an optimized implementation of gradient boosting machines, which sequentially combines weak learners (typically decision trees) to create a strong learner. XGBoost has gained popularity due to its efficiency, scalability, and high performance in various machine learning competitions. XGBoost implements regularization techniques to reduce overfitting and provides several hyperparameters for fine-tuning. However, XGBoost requires careful hyperparameter tuning and can be computationally expensive, especially for large datasets.

Neural Networks Neural Networks (NNs) are a class of models inspired by the structure and function of the human brain. NNs consist of interconnected nodes organized in layers, with each node applying a non-linear activation function to its inputs. NNs are capable of learning complex non-linear relationships in the data and are effective for large datasets with high-dimensional features. However, NNs require a large amount of data to train effectively and are computationally intensive. Additionally, NNs can be prone to overfitting if not regularized properly and can be challenging to interpret due to their complex architecture.

Model Slection

ROC The Receiver Operating Characteristic (ROC) curve is a graphical representation of a binary classification model's performance that illustrates the trade-off between its sensitivity (true positive rate) and specificity (true negative rate) across different threshold values. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings.

AUC The Area Under the ROC Curve (AUC) is a single scalar value that summarizes the overall performance of the model. AUC ranges from 0 to 1, where a higher value indicates better discrimination ability. AUC is a widely used metric for evaluating binary classification models because it provides a comprehensive assessment of the model's performance across all possible threshold settings, making it suitable for comparing different models and assessing their predictive power.

Results:

CA125: This variable has the highest importance based on both metrics (IncMSE and IncNodePurity). A high value indicates that permuting (IncMSE) or using it for splitting nodes (IncNodePurity) leads to a significant increase in prediction error or impurity, suggesting it is a key predictor in your model.

Tumor_size: The second most important variable. While not as influential as CA125, it still has a substantial impact on model performance when its values are permuted or used for node splitting.

Number_of_lymph_nodes: This variable is also important, but to a lesser extent than CA125 and Tumor_size. It still contributes significantly to the model's predictive power.

Interval: This variable has a negative IncMSE value, indicating that permuting it may slightly decrease the mean squared error, but it has a positive IncNodePurity value, suggesting it is still important for node splitting despite the decrease in MSE.

LN_2: This variable has a relatively low importance based on both metrics, indicating it has a smaller impact on model performance compared to the other variables.

Cancer_type_9: This variable has no importance based on both metrics, suggesting it may not be a relevant predictor in your model.

Overall, when interpreting variable importance in a random forest model, it's essential to consider both metrics (IncMSE and IncNodePurity) to get a comprehensive understanding of each variable's impact on the model.

MCC

Accuracy Accuracy (lr_accuracy): It is the proportion of correct predictions over all predictions. In your case, it's approximately 0.966, indicating that the model is accurate in predicting both classes.

Precision Precision (lr_precision): It is the proportion of true positive predictions (metastasis) among all positive predictions (both true positives and false positives). Your precision is 0.5, suggesting that when the model predicts metastasis, it is correct 50% of the time.

Recall Recall (lr_recall), also known as sensitivity or true positive rate (TPR): It is the proportion of true positives predicted correctly among all actual positives. Your recall is 0.571, indicating that the model correctly identifies about 57.1% of actual metastasis cases.

F1 F1 Score (lr_f1): It is the harmonic mean of precision and recall, providing a balance between the two. It is useful when there is an uneven class distribution. Your F1 score is 0.533, suggesting a moderate balance between precision and recall.