```
install.packages("tidyverse")
install.packages("e1071")
install.packages("leaps")
install.packages("tree")
library("tidyverse")
library("e1071")
library("leaps")
library("tree")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘proxy’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message in system("timedatectl", intern = TRUE):
“running command 'timedatectl' had status 1”
── Attaching packages ─────────────────────────────────────
tidyverse 1.3.2 ──
✔ ggplot2 3.4.0      ✔ purrr   0.3.5
✔ tibble  3.1.8      ✔ dplyr   1.0.10
✔ tidyr   1.2.1      ✔ stringr 1.4.1
✔ readr   2.1.3      ✔ forcats 0.5.2
── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

```
install.packages("glmnet")
library("glmnet")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘iterators’, ‘foreach’, ‘shape’,
‘Rcpp’, ‘RcppEigen’


Loading required package: Matrix
```

```
Attaching package: ‘Matrix’


The following objects are masked from ‘package:tidyr’:

    expand, pack, unpack


Loaded glmnet 4.1-6
```

```
install.packages("MASS")
library("MASS")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)


Attaching package: ‘MASS’


The following object is masked from ‘package:dplyr’:

    select
```

```
install.packages("randomForest")
library("randomForest")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

randomForest 4.7-1.1

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


Attaching package: ‘randomForest’


The following object is masked from ‘package:dplyr’:

    combine


The following object is masked from ‘package:ggplot2’:

    margin
```

```
install.packages("kernlab")
library("kernlab")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
Attaching package: ‘kernlab’
```

```
The following object is masked from ‘package:purrr’:

    cross
```

```
The following object is masked from ‘package:ggplot2’:

    alpha
```

```
install.packages("class")
library("class")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
install.packages("pls")
library("pls")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
Attaching package: ‘pls’
```

```
The following object is masked from ‘package:stats’:

    loadings
```

```
heart<-read.csv("heart.csv")
```

```
heart<-heart %>%
mutate(sex=as.factor(sex),cp=as.factor(cp),fbs=as.factor(fbs),restecg=
as.factor(restecg),exang=as.factor(exang),slope=as.factor(slope),ca=as
.factor(ca),thal=as.factor(thal),target=as.factor(target))

heart
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
1	52	1	0	125	212	0	1	168	0	1.0	2	2	3
2	53	1	0	140	203	1	0	155	1	3.1	0	0	3
3	70	1	0	145	174	0	1	125	1	2.6	0	0	3
4	61	1	0	148	203	0	1	161	0	0.0	2	1	3
5	62	0	0	138	294	1	1	106	0	1.9	1	3	2
6	58	0	0	100	248	0	0	122	0	1.0	1	0	2
7	58	1	0	114	318	0	2	140	0	4.4	0	3	1
8	55	1	0	160	289	0	0	145	1	0.8	1	1	3
9	46	1	0	120	249	0	0	144	0	0.8	2	0	3
10	54	1	0	122	286	0	0	116	1	3.2	1	2	2
11	71	0	0	112	149	0	1	125	0	1.6	1	0	2
12	43	0	0	132	341	1	0	136	1	3.0	1	0	3
13	34	0	1	118	210	0	1	192	0	0.7	2	0	2
14	51	1	0	140	298	0	1	122	1	4.2	1	3	3
15	52	1	0	128	204	1	1	156	1	1.0	1	0	0
16	34	0	1	118	210	0	1	192	0	0.7	2	0	2
17	51	0	2	140	308	0	0	142	0	1.5	2	1	2
18	54	1	0	124	266	0	0	109	1	2.2	1	1	3
19	50	0	1	120	244	0	1	162	0	1.1	2	0	2
20	58	1	2	140	211	1	0	165	0	0.0	2	0	2
21	60	1	2	140	185	0	0	155	0	3.0	1	0	2

```
22   67  0   0  106      223  0   1       142     0     0.3     2
2  2
23   45  1   0  104      208  0   0       148     1     3.0     1
0  2
24   63  0   2  135      252  0   0       172     0     0.0     2
0  2
25   42  0   2  120      209  0   1       173     0     0.0     1
0  2
26   61  0   0  145      307  0   0       146     1     1.0     1
0  3
27   44  1   2  130      233  0   1       179     1     0.4     2
0  2
28   58  0   1  136      319  1   0       152     0     0.0     2
2  2
29   56  1   2  130      256  1   0       142     1     0.6     1
1  1
30   55  0   0  180      327  0   2       117     1     3.4     1
0  2
⋮     ⋮   ⋮   ⋮  ⋮          ⋮    ⋮   ⋮        ⋮       ⋮     ⋮
⋮       ⋮   ⋮
996  44  1   1  120      263  0   1       173     0     0.0     2
0  3
997  56  0   0  134      409  0   0       150     1     1.9     1
2  3
998  54  1   0  120      188  0   1       113     0     1.4     1
1  3
999  42  1   0  136      315  0   1       125     1     1.8     1
0  1
1000 67  1   0  125      254  1   1       163     0     0.2     1
2  3
1001 64  1   0  145      212  0   0       132     0     2.0     1
2  1
1002 42  1   0  140      226  0   1       178     0     0.0     2
0  2
1003 66  1   0  112      212  0   0       132     1     0.1     2
1  2
1004 52  1   0  108      233  1   1       147     0     0.1     2
3  3
1005 51  0   2  140      308  0   0       142     0     1.5     2
1  2
1006 55  0   0  128      205  0   2       130     1     2.0     1
1  3
1007 58  1   2  140      211  1   0       165     0     0.0     2
0  2
1008 56  1   3  120      193  0   0       162     0     1.9     1
0  3
1009 42  1   1  120      295  0   1       162     0     0.0     2
0  2
1010 40  1   0  152      223  0   1       181     0     0.0     2
0  3
```

```
1011 51  1   0  140      299  0   1       173     1     1.6     2
0  3
1012 45  1   1  128      308  0   0       170     0     0.0     2
0  2
1013 48  1   1  110      229  0   1       168     0     1.0     0
0  3
1014 58  1   0  114      318  0   2       140     0     4.4     0
3  1
1015 44  0   2  108      141  0   1       175     0     0.6     1
0  2
1016 58  1   0  128      216  0   0       131     1     2.2     1
3  3
1017 65  1   3  138      282  1   0       174     0     1.4     1
1  2
1018 53  1   0  123      282  0   1        95     1     2.0     1
2  3
1019 41  1   0  110      172  0   0       158     0     0.0     2
0  3
1020 47  1   0  112      204  0   1       143     0     0.1     2
0  2
1021 59  1   1  140      221  0   1       164     1     0.0     2
0  2
1022 60  1   0  125      258  0   0       141     1     2.8     1
1  3
1023 47  1   0  110      275  0   0       118     1     1.0     1
1  2
1024 50  0   0  110      254  0   0       159     0     0.0     2
0  2
1025 54  1   0  120      188  0   1       113     0     1.4     1
1  3
```

	target
1	0
2	0
3	0
4	0
5	0
6	1
7	0
8	0
9	0
10	0
11	1
12	0
13	1
14	0
15	0
16	1
17	1
18	0
19	1

```
20   1
21   0
22   1
23   1
24   1
25   1
26   0
27   1
28   0
29   0
30   0
⋮    ⋮
996  1
997  0
998  0
999  0
1000 0
1001 0
1002 1
1003 0
1004 1
1005 1
1006 0
1007 1
1008 1
1009 1
1010 0
1011 0
1012 1
1013 0
1014 0
1015 1
1016 0
1017 0
1018 0
1019 0
1020 1
1021 1
1022 0
1023 0
1024 1
1025 0
```

```
heart %>% filter(target==1)
```

```
    age sex cp trestbps chol fbs restecg thalach exang oldpeak slope
ca thal
1   58  0   0  100      248  0   0       122     0     1.0     1     0
2
2   71  0   0  112      149  0   1       125     0     1.6     1     0
2
```

3 34 0 1 118 210 0 1 192 0 0.7 2 0
2
4 34 0 1 118 210 0 1 192 0 0.7 2 0
2
5 51 0 2 140 308 0 0 142 0 1.5 2 1
2
6 50 0 1 120 244 0 1 162 0 1.1 2 0
2
7 58 1 2 140 211 1 0 165 0 0.0 2 0
2
8 67 0 0 106 223 0 1 142 0 0.3 2 2
2
9 45 1 0 104 208 0 0 148 1 3.0 1 0
2
10 63 0 2 135 252 0 0 172 0 0.0 2 0
2
11 42 0 2 120 209 0 1 173 0 0.0 1 0
2
12 44 1 2 130 233 0 1 179 1 0.4 2 0
2
13 50 0 1 120 244 0 1 162 0 1.1 2 0
2
14 50 1 2 129 196 0 1 163 0 0.0 2 0
2
15 51 1 3 125 213 0 0 125 1 1.4 2 1
2
16 59 1 0 138 271 0 0 182 0 0.0 2 0
2
17 64 1 0 128 263 0 1 105 1 0.2 1 1
3
18 65 0 2 160 360 0 0 151 0 0.8 2 0
2
19 54 1 2 120 258 0 0 147 0 0.4 1 0
3
20 55 0 1 132 342 0 1 166 0 1.2 2 0
2
21 42 1 0 140 226 0 1 178 0 0.0 2 0
2
22 41 1 1 135 203 0 1 132 0 0.0 1 0
1
23 66 0 2 146 278 0 0 152 0 0.0 1 1
2
24 58 0 3 150 283 1 0 162 0 1.0 2 0
2
25 38 1 2 138 175 0 1 173 0 0.0 2 4
2
26 56 1 3 120 193 0 0 162 0 1.9 1 0
3
27 48 1 1 130 245 0 0 180 0 0.2 1 0
2

```
28  29  1   1  130      204  0   0       202     0     0.0     2     0
2
29  66  0   2  146      278  0   0       152     0     0.0     1     1
2
30  59  1   2  150      212  1   1       157     0     1.6     2     0
2
 ⋮   ⋮   ⋮   ⋮  ⋮         ⋮    ⋮   ⋮       ⋮       ⋮     ⋮
 ⋮      ⋮   ⋮
497 44  0   2  108      141  0   1       175     0     0.6     1     0
2
498 76  0   2  140      197  0   2       116     0     1.1     1     0
2
499 60  0   2  120      178  1   1        96     0     0.0     2     0
2
500 52  1   1  120      325  0   1       172     0     0.2     2     0
2
501 38  1   2  138      175  0   1       173     0     0.0     2     4
2
502 52  1   2  172      199  1   1       162     0     0.5     2     0
3
503 52  1   3  118      186  0   0       190     0     0.0     1     0
1
504 51  1   2  125      245  1   0       166     0     2.4     1     0
2
505 43  1   0  110      211  0   1       161     0     0.0     2     0
3
506 52  1   1  128      205  1   1       184     0     0.0     2     0
2
507 57  1   0  140      192  0   1       148     0     0.4     1     0
1
508 67  0   0  106      223  0   1       142     0     0.3     2     2
2
509 64  1   0  128      263  0   1       105     1     0.2     1     1
3
510 59  1   0  135      234  0   1       161     0     0.5     1     0
3
511 62  1   2  130      231  0   1       146     0     1.8     1     3
3
512 71  0   1  160      302  0   1       162     0     0.4     2     2
2
513 56  1   1  120      236  0   1       178     0     0.8     2     0
2
514 50  0   0  110      254  0   0       159     0     0.0     2     0
2
515 44  1   1  120      263  0   1       173     0     0.0     2     0
3
516 42  1   0  140      226  0   1       178     0     0.0     2     0
2
517 52  1   0  108      233  1   1       147     0     0.1     2     3
3
```

```
518 51  0   2  140      308  0   0       142     0     1.5     2     1
2
519 58  1   2  140      211  1   0       165     0     0.0     2     0
2
520 56  1   3  120      193  0   0       162     0     1.9     1     0
3
521 42  1   1  120      295  0   1       162     0     0.0     2     0
2
522 45  1   1  128      308  0   0       170     0     0.0     2     0
2
523 44  0   2  108      141  0   1       175     0     0.6     1     0
2
524 47  1   0  112      204  0   1       143     0     0.1     2     0
2
525 59  1   1  140      221  0   1       164     1     0.0     2     0
2
526 50  0   0  110      254  0   0       159     0     0.0     2     0
2
    target
1   1
2   1
3   1
4   1
5   1
6   1
7   1
8   1
9   1
10  1
11  1
12  1
13  1
14  1
15  1
16  1
17  1
18  1
19  1
20  1
21  1
22  1
23  1
24  1
25  1
26  1
27  1
28  1
29  1
30  1
⋮    ⋮
```

```
497 1
498 1
499 1
500 1
501 1
502 1
503 1
504 1
505 1
506 1
507 1
508 1
509 1
510 1
511 1
512 1
513 1
514 1
515 1
516 1
517 1
518 1
519 1
520 1
521 1
522 1
523 1
524 1
525 1
526 1
```

```
heart %>% filter(target==0)
```

```
    age sex cp trestbps chol fbs restecg thalach exang oldpeak slope
ca thal
1   52  1   0  125      212  0   1       168     0     1.0     2     2
3
2   53  1   0  140      203  1   0       155     1     3.1     0     0
3
3   70  1   0  145      174  0   1       125     1     2.6     0     0
3
4   61  1   0  148      203  0   1       161     0     0.0     2     1
3
5   62  0   0  138      294  1   1       106     0     1.9     1     3
2
6   58  1   0  114      318  0   2       140     0     4.4     0     3
1
7   55  1   0  160      289  0   0       145     1     0.8     1     1
3
8   46  1   0  120      249  0   0       144     0     0.8     2     0
3
```

9	54	1	0	122	286	0	0	116	1	3.2	1	2	2
10	43	0	0	132	341	1	0	136	1	3.0	1	0	3
11	51	1	0	140	298	0	1	122	1	4.2	1	3	3
12	52	1	0	128	204	1	1	156	1	1.0	1	0	0
13	54	1	0	124	266	0	0	109	1	2.2	1	1	3
14	60	1	2	140	185	0	0	155	0	3.0	1	0	2
15	61	0	0	145	307	0	0	146	1	1.0	1	0	3
16	58	0	1	136	319	1	0	152	0	0.0	2	2	2
17	56	1	2	130	256	1	0	142	1	0.6	1	1	1
18	55	0	0	180	327	0	2	117	1	3.4	1	0	2
19	44	1	0	120	169	0	1	144	1	2.8	0	0	1
20	57	1	0	130	131	0	1	115	1	1.2	1	1	3
21	70	1	2	160	269	0	1	112	1	2.9	1	1	3
22	46	1	2	150	231	0	1	147	0	3.6	1	0	2
23	57	1	2	128	229	0	0	150	0	0.4	1	1	3
24	61	0	0	130	330	0	0	169	0	0.0	2	0	2
25	46	1	0	120	249	0	0	144	0	0.8	2	0	3
26	66	0	0	178	228	1	1	165	1	1.0	1	2	3
27	60	1	0	117	230	1	1	160	1	1.4	2	2	3
28	57	0	0	140	241	0	1	123	1	0.2	1	0	3
29	49	1	2	120	188	0	1	139	0	2.0	1	3	3
30	55	1	0	140	217	0	1	111	1	5.6	0	0	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
470	53	1	0	140	203	1	0	155	1	3.1	0	0	3
471	39	1	0	118	219	0	1	140	0	1.2	1	0	3

```
472 63  0   0  108      269  0   1        169     1     1.8     1     2
2
473 44  1   0  110      197  0   0        177     0     0.0     2     1
2
474 45  1   0  142      309  0   0        147     1     0.0     1     3
3
475 39  1   0  118      219  0   1        140     0     1.2     1     0
3
476 55  0   0  180      327  0   2        117     1     3.4     1     0
2
477 57  1   1  154      232  0   0        164     0     0.0     2     1
2
478 60  1   0  140      293  0   0        170     0     1.2     1     2
3
479 60  1   0  117      230  1   1        160     1     1.4     2     2
3
480 43  1   0  132      247  1   0        143     1     0.1     1     4
3
481 59  1   0  110      239  0   0        142     1     1.2     1     1
3
482 56  0   0  134      409  0   0        150     1     1.9     1     2
3
483 54  1   0  120      188  0   1        113     0     1.4     1     1
3
484 42  1   0  136      315  0   1        125     1     1.8     1     0
1
485 67  1   0  125      254  1   1        163     0     0.2     1     2
3
486 64  1   0  145      212  0   0        132     0     2.0     1     2
1
487 66  1   0  112      212  0   0        132     1     0.1     2     1
2
488 55  0   0  128      205  0   2        130     1     2.0     1     1
3
489 40  1   0  152      223  0   1        181     0     0.0     2     0
3
490 51  1   0  140      299  0   1        173     1     1.6     2     0
3
491 48  1   1  110      229  0   1        168     0     1.0     0     0
3
492 58  1   0  114      318  0   2        140     0     4.4     0     3
1
493 58  1   0  128      216  0   0        131     1     2.2     1     3
3
494 65  1   3  138      282  1   0        174     0     1.4     1     1
2
495 53  1   0  123      282  0   1         95     1     2.0     1     2
3
496 41  1   0  110      172  0   0        158     0     0.0     2     0
3
```

```
497 60  1   0  125      258  0   0       141     1     2.8     1     1
3
498 47  1   0  110      275  0   0       118     1     1.0     1     1
2
499 54  1   0  120      188  0   1       113     0     1.4     1     1
3
    target
1   0
2   0
3   0
4   0
5   0
6   0
7   0
8   0
9   0
10  0
11  0
12  0
13  0
14  0
15  0
16  0
17  0
18  0
19  0
20  0
21  0
22  0
23  0
24  0
25  0
26  0
27  0
28  0
29  0
30  0
⋮    ⋮
470 0
471 0
472 0
473 0
474 0
475 0
476 0
477 0
478 0
479 0
480 0
481 0
```

```
482 0
483 0
484 0
485 0
486 0
487 0
488 0
489 0
490 0
491 0
492 0
493 0
494 0
495 0
496 0
497 0
498 0
499 0

summary(heart)

      age        sex     cp          trestbps          chol     fbs
restecg
 Min.   :29.00   0:312   0:497   Min.   : 94.0   Min.   :126   0:872
0:497
 1st Qu.:48.00   1:713   1:167   1st Qu.:120.0   1st Qu.:211   1:153
1:513
 Median :56.00           2:284   Median :130.0   Median :240
2: 15
 Mean   :54.43           3: 77   Mean   :131.6   Mean   :246

 3rd Qu.:61.00                   3rd Qu.:140.0   3rd Qu.:275

 Max.   :77.00                   Max.   :200.0   Max.   :564

    thalach      exang      oldpeak      slope   ca      thal
target
 Min.   : 71.0   0:680   Min.   :0.000   0: 74   0:578   0:  7   0:499

 1st Qu.:132.0   1:345   1st Qu.:0.000   1:482   1:226   1: 64   1:526

 Median :152.0           Median :0.800   2:469   2:134   2:544

 Mean   :149.1           Mean   :1.072           3: 69   3:410

 3rd Qu.:166.0           3rd Qu.:1.800           4: 18

 Max.   :202.0           Max.   :6.200
```

```r
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
heartmm <- as.data.frame(model.matrix(~.-1,heart))
# we are going to normalize everything
heart <- as.data.frame(lapply(heartmm, normalize))

set.seed(385720)

train<-sample(dim(heart)[1],dim(heart)[1]*0.7)

test<--train

heart.train<-heart[train,]

heart.test<-heart[test,]

head(heart)
```

```
  age       sex0 sex1 cp1 cp2 cp3 trestbps   chol      fbs1 restecg1 …
slope1
1 0.4791667 0    1    0   0   0   0.29245283 0.1963470 0    1         …
0
2 0.5000000 0    1    0   0   0   0.43396226 0.1757991 1    0         …
0
3 0.8541667 0    1    0   0   0   0.48113208 0.1095890 0    1         …
0
4 0.6666667 0    1    0   0   0   0.50943396 0.1757991 0    1         …
0
5 0.6875000 1    0    0   0   0   0.41509434 0.3835616 1    1         …
1
6 0.6041667 1    0    0   0   0   0.05660377 0.2785388 0    0         …
1
  slope2 ca1 ca2 ca3 ca4 thal1 thal2 thal3 target1
1 1      0   1   0   0   0     0     1     0
2 0      0   0   0   0   0     0     1     0
3 0      0   0   0   0   0     0     1     0
4 1      1   0   0   0   0     0     1     0
5 0      0   0   1   0   0     1     0     0
6 0      0   0   0   0   0     1     0     1
```

Logistic Regression

```r
logmod <- glm(target1 ~ ., data = heart.train, family = "binomial")
summary(logmod)
logmodselect =  step(logmod, direction = "backward")
```

```
Call:
glm(formula = target1 ~ ., family = "binomial", data = heart.train)

Deviance Residuals:
```

```
     Min        1Q    Median        3Q       Max
-2.87521  -0.29896   0.07339   0.41120   3.10889

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.0625     2.6834  -0.396 0.692141
age           1.0789     0.8296   1.300 0.193439
sex0          2.0465     0.3846   5.320 1.04e-07 ***
sex1              NA         NA      NA       NA
cp1           0.8673     0.3885   2.232 0.025591 *
cp2           1.8740     0.3457   5.421 5.93e-08 ***
cp3           2.6358     0.4723   5.580 2.40e-08 ***
trestbps     -2.9214     0.8638  -3.382 0.000719 ***
chol         -3.1810     1.2239  -2.599 0.009346 **
fbs1          0.1954     0.4047   0.483 0.629268
restecg1      0.2423     0.2727   0.889 0.374227
restecg2     -0.4054     1.8375  -0.221 0.825385
thalach       2.4574     1.0658   2.306 0.021123 *
exang1       -0.7879     0.3101  -2.541 0.011064 *
oldpeak      -3.6010     1.0194  -3.532 0.000412 ***
slope1       -0.9724     0.5796  -1.678 0.093416 .
slope2        0.4652     0.6165   0.755 0.450470
ca1          -2.2040     0.3556  -6.198 5.73e-10 ***
ca2          -3.4790     0.5198  -6.693 2.19e-11 ***
ca3          -2.3177     0.6775  -3.421 0.000624 ***
ca4           0.8779     1.3773   0.637 0.523873
thal1         2.1875     2.5646   0.853 0.393678
thal2         1.9098     2.5170   0.759 0.447994
thal3         0.5836     2.5190   0.232 0.816801
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 993.66  on 716  degrees of freedom
Residual deviance: 409.76  on 694  degrees of freedom
AIC: 455.76

Number of Fisher Scoring iterations: 6

Start:  AIC=455.76
target1 ~ age + sex0 + sex1 + cp1 + cp2 + cp3 + trestbps + chol +
    fbs1 + restecg1 + restecg2 + thalach + exang1 + oldpeak +
    slope1 + slope2 + ca1 + ca2 + ca3 + ca4 + thal1 + thal2 +
    thal3


Step:  AIC=455.76
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + fbs1 +
    restecg1 + restecg2 + thalach + exang1 + oldpeak + slope1 +
```

```
    slope2 + ca1 + ca2 + ca3 + ca4 + thal1 + thal2 + thal3

           Df Deviance    AIC
- restecg2  1   409.81 453.81
- thal3     1   409.82 453.82
- fbs1      1   410.00 454.00
- ca4       1   410.17 454.17
- slope2    1   410.32 454.32
- thal2     1   410.39 454.39
- restecg1  1   410.56 454.56
- thal1     1   410.59 454.59
- age       1   411.47 455.47
<none>          409.76 455.76
- slope1    1   412.64 456.64
- cp1       1   414.91 458.91
- thalach   1   415.35 459.35
- exang1    1   416.20 460.20
- chol      1   416.48 460.48
- trestbps  1   421.78 465.78
- oldpeak   1   423.53 467.53
- ca3       1   424.73 468.73
- cp2       1   441.93 485.93
- sex0      1   442.06 486.06
- cp3       1   445.30 489.30
- ca1       1   452.81 496.81
- ca2       1   464.99 508.99

Step:  AIC=453.81
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + fbs1 +
    restecg1 + thalach + exang1 + oldpeak + slope1 + slope2 +
    ca1 + ca2 + ca3 + ca4 + thal1 + thal2 + thal3

           Df Deviance    AIC
- thal3     1   409.87 451.87
- fbs1      1   410.05 452.05
- ca4       1   410.22 452.22
- slope2    1   410.37 452.37
- thal2     1   410.44 452.44
- thal1     1   410.63 452.63
- restecg1  1   410.65 452.65
- age       1   411.50 453.50
<none>          409.81 453.81
- slope1    1   412.68 454.68
- cp1       1   414.97 456.97
- thalach   1   415.43 457.43
- exang1    1   416.23 458.23
- chol      1   416.48 458.48
- trestbps  1   421.85 463.85
- oldpeak   1   423.91 465.91
- ca3       1   424.92 466.92
```

```
- cp2       1   442.00 484.00
- sex0      1   442.18 484.18
- cp3       1   445.46 487.46
- ca1       1   452.91 494.91
- ca2       1   465.01 507.01

Step:  AIC=451.87
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + fbs1 +
    restecg1 + thalach + exang1 + oldpeak + slope1 + slope2 +
    ca1 + ca2 + ca3 + ca4 + thal1 + thal2

           Df Deviance    AIC
- fbs1      1   410.09 450.09
- ca4       1   410.28 450.28
- slope2    1   410.42 450.42
- restecg1  1   410.70 450.70
- age       1   411.56 451.56
<none>          409.87 451.87
- slope1    1   412.74 452.74
- cp1       1   415.05 455.05
- thalach   1   415.50 455.50
- exang1    1   416.31 456.31
- chol      1   416.49 456.49
- thal1     1   416.97 456.97
- trestbps  1   421.87 461.87
- oldpeak   1   423.93 463.93
- ca3       1   424.93 464.93
- thal2     1   431.44 471.44
- cp2       1   442.02 482.02
- sex0      1   442.23 482.23
- cp3       1   445.55 485.55
- ca1       1   452.97 492.97
- ca2       1   465.02 505.02

Step:  AIC=450.09
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + restecg1 +
    thalach + exang1 + oldpeak + slope1 + slope2 + ca1 + ca2 +
    ca3 + ca4 + thal1 + thal2

           Df Deviance    AIC
- ca4       1   410.56 448.56
- slope2    1   410.61 448.61
- restecg1  1   410.90 448.90
- age       1   411.75 449.75
<none>          410.09 450.09
- slope1    1   413.02 451.02
- cp1       1   415.56 453.56
- thalach   1   415.84 453.84
- exang1    1   416.36 454.36
- chol      1   416.58 454.58
```

```
- thal1     1   417.90 455.90
- trestbps  1   421.87 459.87
- oldpeak   1   424.62 462.62
- ca3       1   424.93 462.93
- thal2     1   431.77 469.77
- sex0      1   442.26 480.26
- cp2       1   444.41 482.41
- cp3       1   446.62 484.62
- ca1       1   453.18 491.18
- ca2       1   465.25 503.25

Step:  AIC=448.56
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + restecg1 +
    thalach + exang1 + oldpeak + slope1 + slope2 + ca1 + ca2 +
    ca3 + thal1 + thal2

           Df Deviance    AIC
- slope2    1   411.05 447.05
- restecg1  1   411.39 447.39
- age       1   412.12 448.12
<none>          410.56 448.56
- slope1    1   413.43 449.43
- thalach   1   416.19 452.19
- cp1       1   416.30 452.30
- exang1    1   416.74 452.74
- chol      1   417.12 453.12
- thal1     1   418.22 454.22
- trestbps  1   422.24 458.24
- ca3       1   425.54 461.54
- oldpeak   1   425.66 461.66
- thal2     1   431.98 467.98
- sex0      1   442.36 478.36
- cp2       1   445.01 481.01
- cp3       1   446.88 482.88
- ca1       1   454.06 490.06
- ca2       1   465.46 501.46

Step:  AIC=447.05
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + restecg1 +
    thalach + exang1 + oldpeak + slope1 + ca1 + ca2 + ca3 + thal1 +
    thal2

           Df Deviance    AIC
- restecg1  1   412.03 446.03
- age       1   412.59 446.59
<none>          411.05 447.05
- cp1       1   416.78 450.78
- thalach   1   416.88 450.88
- exang1    1   417.20 451.20
- chol      1   417.57 451.57
```

```
- thal1     1   418.31 452.31
- trestbps  1   422.61 456.61
- ca3       1   425.76 459.76
- slope1    1   430.45 464.45
- thal2     1   432.47 466.47
- oldpeak   1   433.78 467.78
- sex0      1   442.66 476.66
- cp2       1   445.52 479.52
- cp3       1   447.47 481.47
- ca1       1   454.11 488.11
- ca2       1   465.58 499.58

Step:  AIC=446.03
target1 ~ age + sex0 + cp1 + cp2 + cp3 + trestbps + chol + thalach +
    exang1 + oldpeak + slope1 + ca1 + ca2 + ca3 + thal1 + thal2

           Df Deviance    AIC
- age       1   413.43 445.43
<none>          412.03 446.03
- cp1       1   417.64 449.64
- thalach   1   417.87 449.87
- exang1    1   418.14 450.14
- thal1     1   419.23 451.23
- chol      1   420.02 452.02
- trestbps  1   424.15 456.15
- ca3       1   427.32 459.32
- thal2     1   432.61 464.61
- slope1    1   433.14 465.14
- oldpeak   1   434.60 466.60
- sex0      1   445.27 477.27
- cp2       1   446.44 478.44
- cp3       1   447.66 479.66
- ca1       1   457.43 489.43
- ca2       1   466.51 498.51

Step:  AIC=445.43
target1 ~ sex0 + cp1 + cp2 + cp3 + trestbps + chol + thalach +
    exang1 + oldpeak + slope1 + ca1 + ca2 + ca3 + thal1 + thal2

           Df Deviance    AIC
<none>          413.43 445.43
- thalach   1   417.94 447.94
- cp1       1   419.53 449.53
- exang1    1   419.92 449.92
- chol      1   420.53 450.53
- thal1     1   421.01 451.01
- trestbps  1   424.23 454.23
- ca3       1   427.71 457.71
- slope1    1   433.69 463.69
- thal2     1   433.87 463.87
```

```
- oldpeak   1   437.45 467.45
- sex0      1   447.91 477.91
- cp2       1   448.62 478.62
- cp3       1   450.00 480.00
- ca1       1   457.43 487.43
- ca2       1   467.52 497.52
```

```
colnames(heart)
```

```
 [1] "age"      "sex0"     "sex1"     "cp1"      "cp2"      "cp3"
 [7] "trestbps" "chol"     "fbs1"     "restecg1" "restecg2" "thalach"
[13] "exang1"   "oldpeak"  "slope1"   "slope2"   "ca1"      "ca2"
[19] "ca3"      "ca4"      "thal1"    "thal2"    "thal3"    "target1"
```

```
bestmod <- glm(target1 ~ sex0 + cp1 + cp2 + cp3 + trestbps + chol +
thalach +
    exang1 + oldpeak + slope1 + ca1 + ca2 + ca3 + thal1 + thal2, data
= heart.train, family = "binomial")

logpred = predict(bestmod, newdata = heart.test, type = "response")
logpred = ifelse(logpred > 0.5, 1, 0)
table(logpred, heart.test$target1)
logRegaccuracy<-mean(logpred == heart.test$target1)
```

```
logpred   0   1
      0 119  21
      1  29 139
```

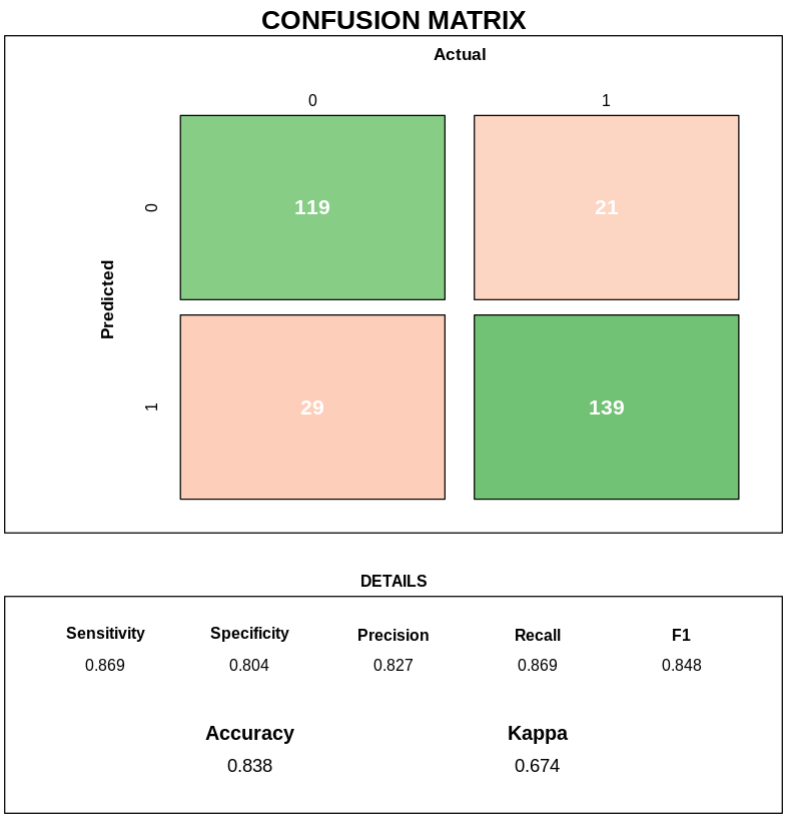
```
install.packages("caret")
library(caret)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘listenv’, ‘parallelly’, ‘future’,
‘globals’, ‘future.apply’, ‘numDeriv’, ‘progressr’, ‘SQUAREM’, ‘lava’,
‘prodlim’, ‘clock’, ‘gower’, ‘hardhat’, ‘ipred’, ‘timeDate’,
‘ModelMetrics’, ‘plyr’, ‘pROC’, ‘recipes’, ‘reshape2’


Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:pls’:

    R2
```

```
The following object is masked from ‘package:purrr’:

    lift
```

```
cm <- confusionMatrix(data = as.factor(logpred), reference =
as.factor(heart.test$target1),positive='1')

draw_confusion_matrix <- function(cm) {

  total <- sum(cm$table)
  res <- as.numeric(cm$table)

  # Generate color gradients. Palettes come from RColorBrewer.
  greenPalette <-
c("#F7FCF5","#E5F5E0","#C7E9C0","#A1D99B","#74C476","#41AB5D","#238B45
","#006D2C","#00441B")
  redPalette <-
c("#FFF5F0","#FEE0D2","#FCBBA1","#FC9272","#FB6A4A","#EF3B2C","#CB181D
","#A50F15","#67000D")
  getColor <- function (greenOrRed = "green", amount = 0) {
    if (amount == 0)
      return("#FFFFFF")
    palette <- greenPalette
    if (greenOrRed == "red")
      palette <- redPalette
    colorRampPalette(palette)(100)[10 + ceiling(90 * amount / total)]
  }

  # set the basic layout
  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="",
xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  classes = colnames(cm$table)
  rect(150, 430, 240, 370, col=getColor("green", res[1]))
  text(195, 435, classes[1], cex=1.2)
  rect(250, 430, 340, 370, col=getColor("red", res[3]))
  text(295, 435, classes[2], cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col=getColor("red", res[2]))
  rect(250, 305, 340, 365, col=getColor("green", res[4]))
  text(140, 400, classes[1], cex=1.2, srt=90)
  text(140, 335, classes[2], cex=1.2, srt=90)
```

```r
  # add in the cm results
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main =
"DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}

draw_confusion_matrix(cm)
```




```
logRegaccuracy
```

```
[1] 0.8376623
```

Linear Discriminant Analysis

```
lda.fit <- lda(target1 ~ ., data = heart.train)
```

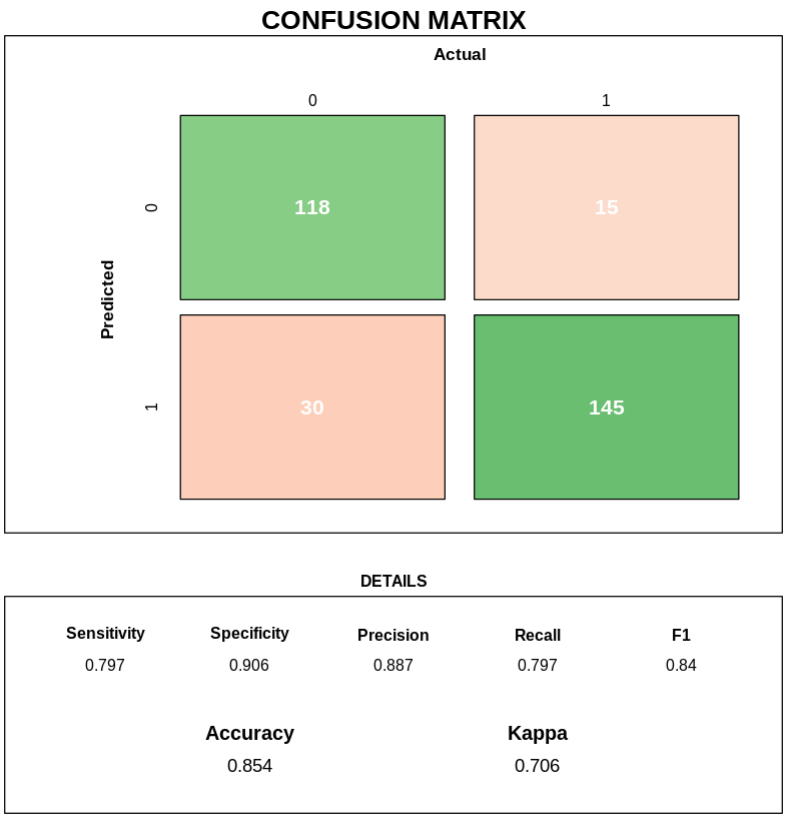
```
Warning message in lda.default(x, grouping, ...):
“variables are collinear”
```

```
lda.pred <- predict(lda.fit, heart.test)
names(lda.pred)
```

```
[1] "class"     "posterior" "x"
```

```
lda.class <- lda.pred$class
cm<-confusionMatrix(as.factor(lda.class),
as.factor(heart.test$target1))
```

```
draw_confusion_matrix(cm)
```




```
ldaaccuracy<-mean(lda.class==heart.test$target1)
ldaaccuracy
```

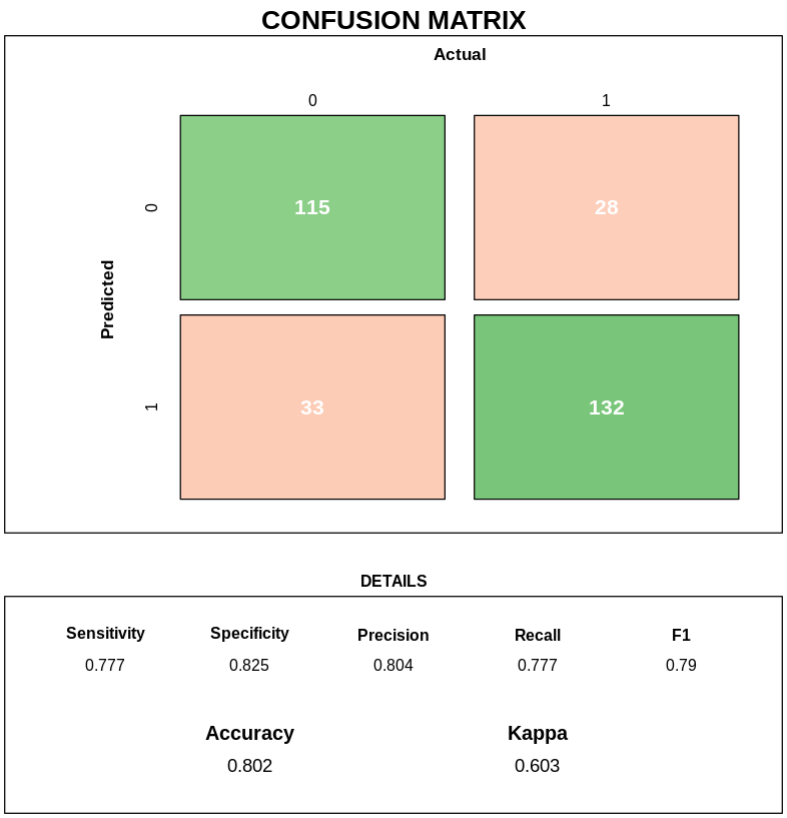
```
[1] 0.8538961
```

Naive Bayes

```
nb.fit <- naiveBayes(target1~., data = heart.train)
```

```
nb.class <- predict(nb.fit, heart.test)
cm<-confusionMatrix(as.factor(nb.class),
as.factor(heart.test$target1))
```

```
draw_confusion_matrix(cm)
```




```
naivebayesaccuracy<-mean(nb.class==heart.test$target1)
naivebayesaccuracy
```

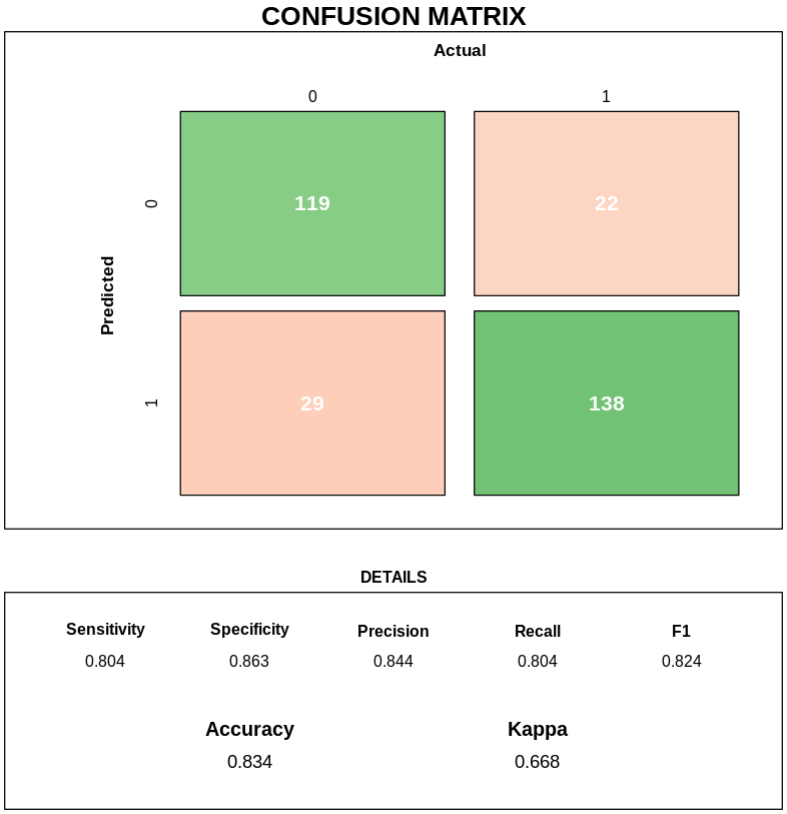
```
[1] 0.8019481
```

SVM

```
svm.linear <- tune(svm, target1 ~ ., data = heart.train, kernel =
"linear",
    ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)))

svm.linear$best.model
```

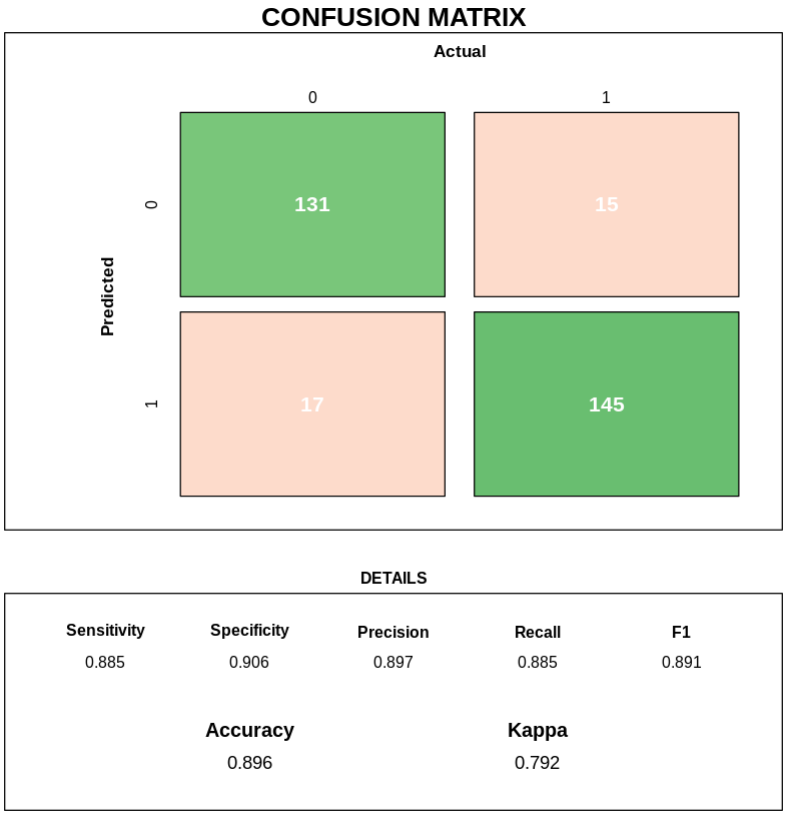
```
Call:
best.tune(METHOD = svm, train.x = target1 ~ ., data = heart.train,
    ranges = list(cost = c(0.001, 0.01, 0.1, 1, 5, 10, 100)), kernel =
"linear")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  0.01
      gamma:  0.04347826
    epsilon:  0.1
```

```
Number of Support Vectors:  576

pred.linear<-predict(svm.linear$best.model,heart.test)
pred.linear<-ifelse(pred.linear>0.5,1,0)

cm<-
confusionMatrix(data=as.factor(pred.linear),reference=as.factor(heart.
test$target1))

draw_confusion_matrix(cm)
```




```
linearsvmaccuracy<-mean(pred.linear==heart.test$target1)

linearsvmaccuracy

[1] 0.8344156

svm.radial <- ksvm(target1 ~ ., data = heart.train, kernel = "rbfdot")
pred.radial <- predict(svm.radial, newdata = heart.test)
pred.radial <- ifelse(pred.radial > 0.5, 1, 0)
cm<-
confusionMatrix(data=as.factor(pred.radial),reference=as.factor(heart.
test$target1))
```

```
draw_confusion_matrix(cm)
```



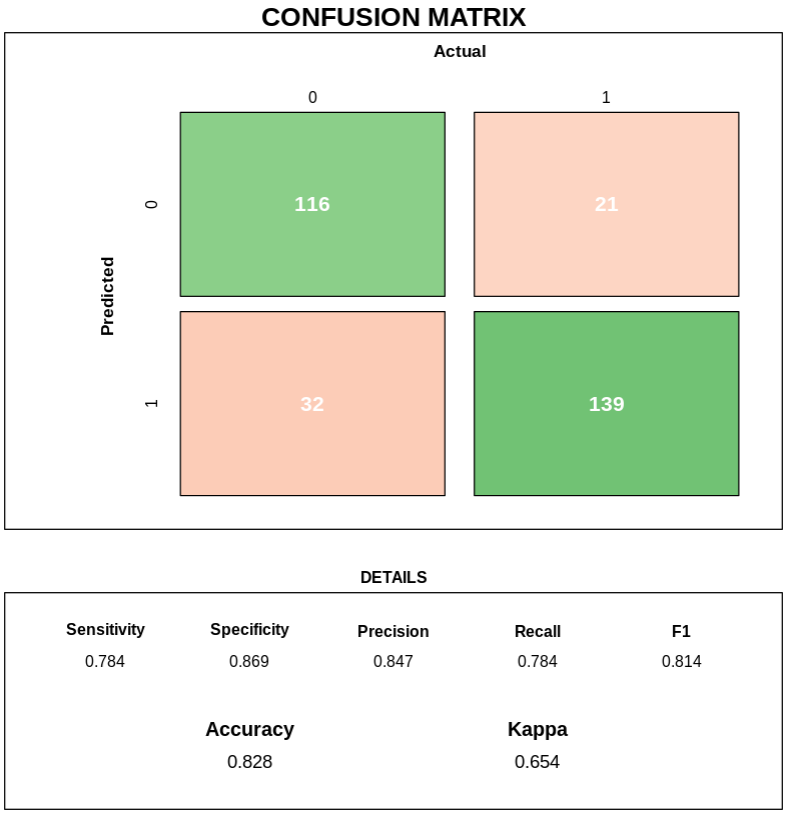

```
radialsvmaccuracy<-mean(pred.radial==heart.test$target1)

radialsvmaccuracy

[1] 0.8961039
```

KNN

```
knntrain <- heart.train
knntesting <- heart.test
trainlabels <- knntrain$target1
testing_labels <- knntesting$target1
knntrain$target1 <- NULL
knntesting$target1 <- NULL
knnprediction <- knn(train = knntrain, test = knntesting,
                     cl = trainlabels, k=14)

knnprediction

  [1] 0 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 1 0 1 1
1 1 0 1 0
 [38] 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 1 0 1
0 0 1 1 1
 [75] 0 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 0 1 1 1 0 1 1 1 1
```

```
1 0 0 1 1
[112] 1 1 1 0 0 0 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0
0 1 1 1 1
[149] 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 0 1 1 0 1 0 0 0
1 1 0 0 1
[186] 1 0 1 0 1 1 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0
1 1 0 1 1
[223] 1 1 0 1 1 1 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1
1 1 1 1 1
[260] 1 1 0 0 0 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1
1 1 1 0 0
[297] 1 1 1 0 0 0 0 0 1 0 0 1
Levels: 0 1
```

```
cm<-confusionMatrix(as.factor(knnprediction),
as.factor(heart.test$target1))

draw_confusion_matrix(cm)
```




```
knnaccuracy<-mean(knnprediction==heart.test$target1)
knnaccuracy

[1] 0.8279221

accuracy<-list()
for(i in 1:50)
```

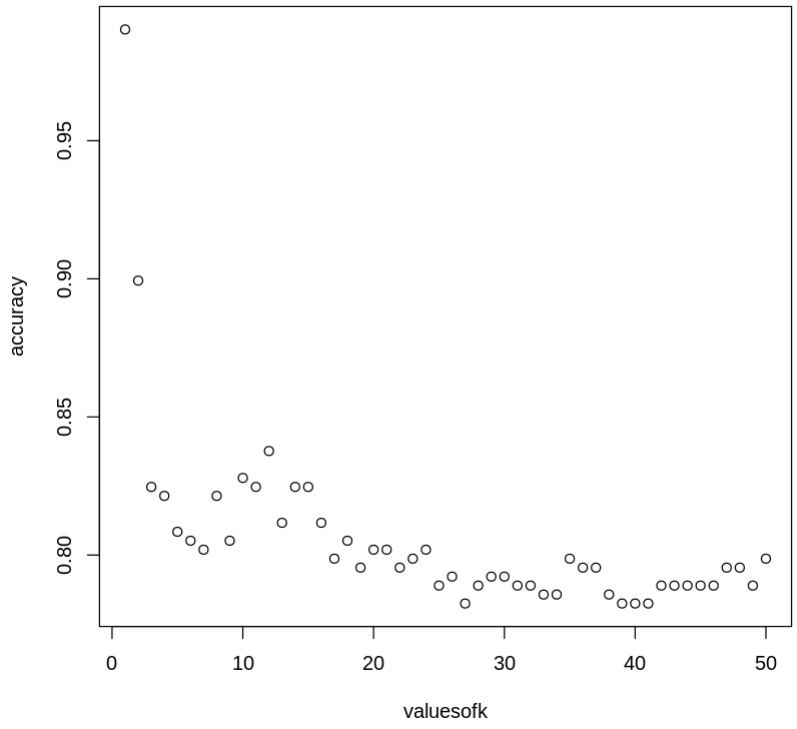
```
{
  knntrain<-heart.train
  knntesting<-heart.test
  trainlabels<-knntrain$target1
  testing_labels<-knntesting$target1
  knntrain$target1<-NULL
  knntesting$target1<-NULL
  knnprediction<-
knn(train=knntrain,test=knntesting,cl=trainlabels,k=i)
  knnaccuracy<-mean(knnprediction==heart.test$target1)
  accuracy<-append(accuracy,knnaccuracy)
}

valuesofk<-1:50

accuracy

[[1]]
[1] 0.9902597

[[2]]
[1] 0.8993506

[[3]]
[1] 0.8246753

[[4]]
[1] 0.8214286

[[5]]
[1] 0.8084416

[[6]]
[1] 0.8051948

[[7]]
[1] 0.8019481

[[8]]
[1] 0.8214286

[[9]]
[1] 0.8051948

[[10]]
[1] 0.8279221

[[11]]
[1] 0.8246753
```

```
[[12]]
[1] 0.8376623

[[13]]
[1] 0.8116883

[[14]]
[1] 0.8246753

[[15]]
[1] 0.8246753

[[16]]
[1] 0.8116883

[[17]]
[1] 0.7987013

[[18]]
[1] 0.8051948

[[19]]
[1] 0.7954545

[[20]]
[1] 0.8019481

[[21]]
[1] 0.8019481

[[22]]
[1] 0.7954545

[[23]]
[1] 0.7987013

[[24]]
[1] 0.8019481

[[25]]
[1] 0.788961

[[26]]
[1] 0.7922078

[[27]]
[1] 0.7824675

[[28]]
```

```
[1] 0.788961

[[29]]
[1] 0.7922078

[[30]]
[1] 0.7922078

[[31]]
[1] 0.788961

[[32]]
[1] 0.788961

[[33]]
[1] 0.7857143

[[34]]
[1] 0.7857143

[[35]]
[1] 0.7987013

[[36]]
[1] 0.7954545

[[37]]
[1] 0.7954545

[[38]]
[1] 0.7857143

[[39]]
[1] 0.7824675

[[40]]
[1] 0.7824675

[[41]]
[1] 0.7824675

[[42]]
[1] 0.788961

[[43]]
[1] 0.788961

[[44]]
[1] 0.788961
```

```
[[45]]
[1] 0.788961

[[46]]
[1] 0.788961

[[47]]
[1] 0.7954545

[[48]]
[1] 0.7954545

[[49]]
[1] 0.788961

[[50]]
[1] 0.7987013
```

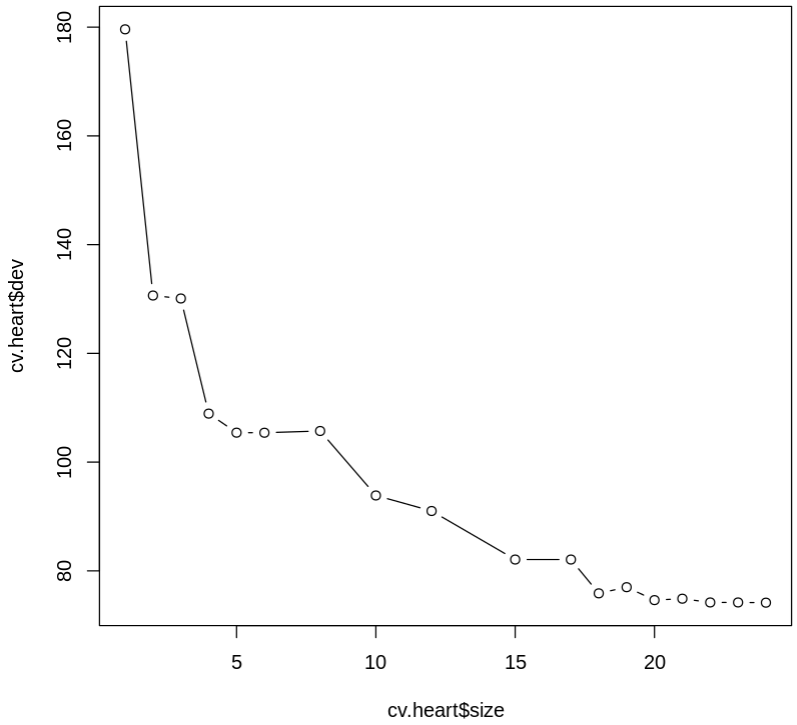
```
plot(valuesofk,accuracy)
```

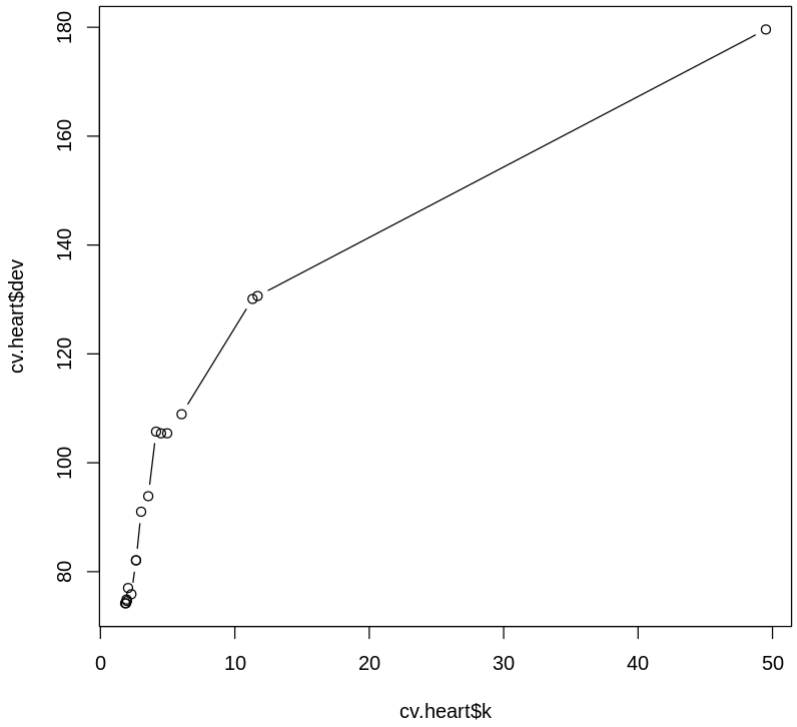


Tree

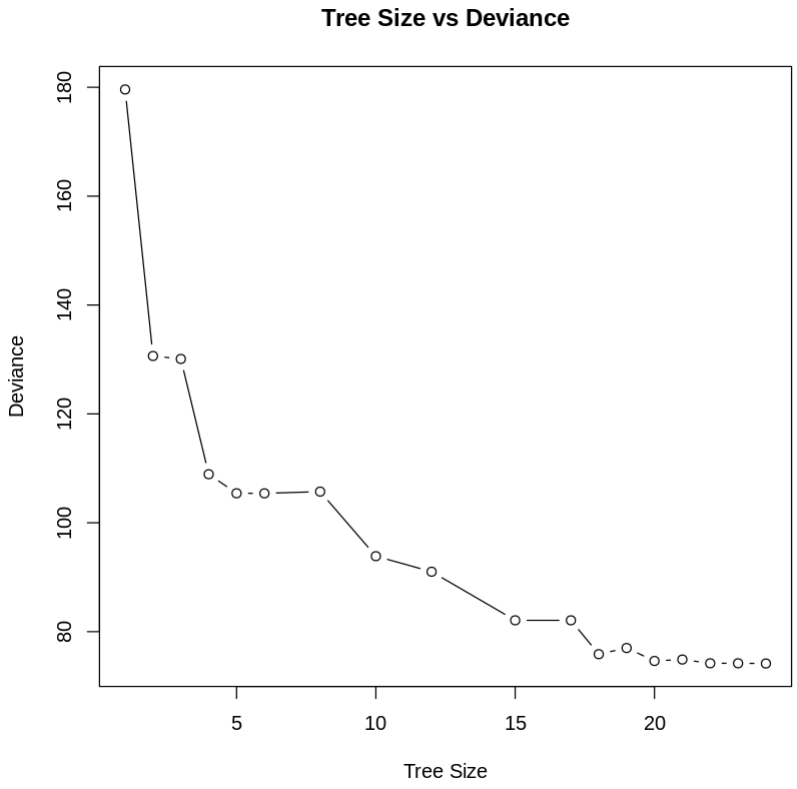
```
install.packages("gbm")
library(gbm)
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Loaded gbm 2.1.8.1


heart.tree<-tree(target1~.,data=heart.train)
summary(heart.tree)


Regression tree:
tree(formula = target1 ~ ., data = heart.train)
Variables actually used in tree construction:
 [1] "thal2"    "oldpeak"  "trestbps" "chol"     "restecg1" "age"
 [7] "cp3"      "cp2"      "thalach"  "exang1"   "ca3"      "slope1"
Number of terminal nodes:  24
Residual mean deviance:  0.06084 = 42.16 / 693
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.91410 -0.02235  0.00000  0.00000  0.08594  0.97770

cv.heart <- cv.tree(heart.tree, FUN = prune.tree,K=10)

plot(cv.heart$size, cv.heart$dev, type = "b")
plot(cv.heart$k, cv.heart$dev, type = "b")
```

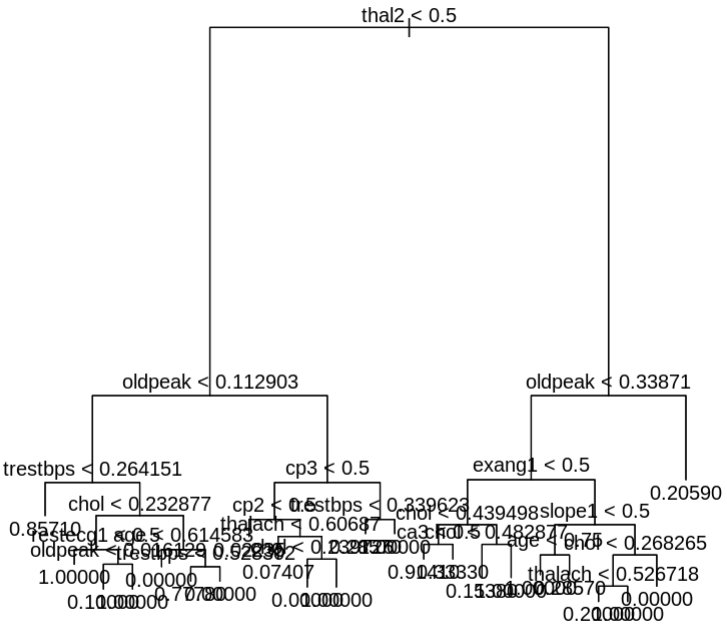




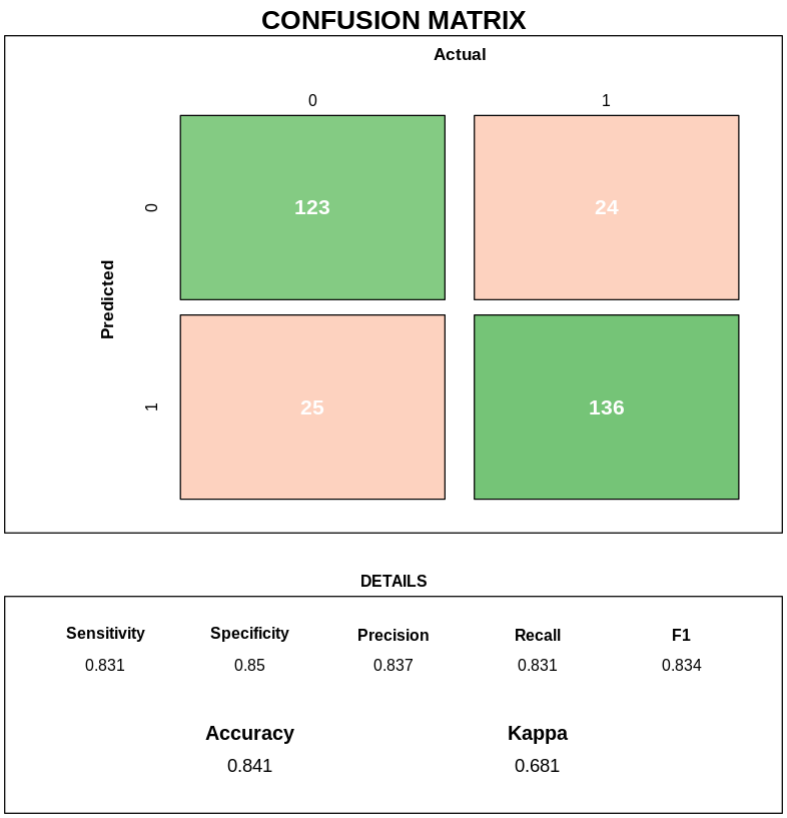

```
plot(cv.heart$size, cv.heart$dev, type = "b", xlab = "Tree Size", ylab
= "Deviance", main = "Tree Size vs Deviance")
```




```
cv.heart$size

 [1] 24 23 22 21 20 19 18 17 15 12 10  8  6  5  4  3  2  1

cv.heart$dev

 [1]  74.15202  74.19153  74.19153  74.88207  74.62077  76.99959
75.86681
 [8]  82.08400  82.08400  91.00989  93.86512 105.72492 105.40436
105.41887
[15] 108.92160 130.09359 130.63983 179.59392

heart.pruned<-prune.tree(heart.tree,best=23)

plot(heart.pruned)
text(heart.pruned,pretty=0)
```




```
summary(heart.pruned)


Regression tree:
snip.tree(tree = heart.tree, nodes = 7L)
Variables actually used in tree construction:
 [1] "thal2"    "oldpeak"  "trestbps" "chol"     "restecg1" "age"
 [7] "cp3"      "cp2"      "thalach"  "exang1"   "ca3"      "slope1"
Number of terminal nodes:  23
Residual mean deviance:  0.0634 = 44 / 694
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.91410 -0.02235  0.00000  0.00000  0.08594  0.97770

predheart.pruned<-predict(heart.pruned,heart.test)

predheart.pruned<-ifelse(predheart.pruned>0.5,1,0)

cm<-
confusionMatrix(data=as.factor(predheart.pruned),reference=as.factor(h
eart.test$target1))

draw_confusion_matrix(cm)
```




```
table(predheart.pruned,heart.test$target1)
```

```
predheart.pruned   0   1
               0 123  24
               1  25 136
```

```
treeaccuracy<-mean(predheart.pruned==heart.test$target1)
treeaccuracy
```

```
[1] 0.8409091
```

Random Forest

```
install.packages("randomForest")
library(randomForest)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
bag.heart<-randomForest(target1~.,data = heart.train,mtry = 5,ntree =
100,importance = T)
```

```
Warning message in randomForest.default(m, y, ...):
“The response has five or fewer unique values.  Are you sure you want
to do regression?”
```

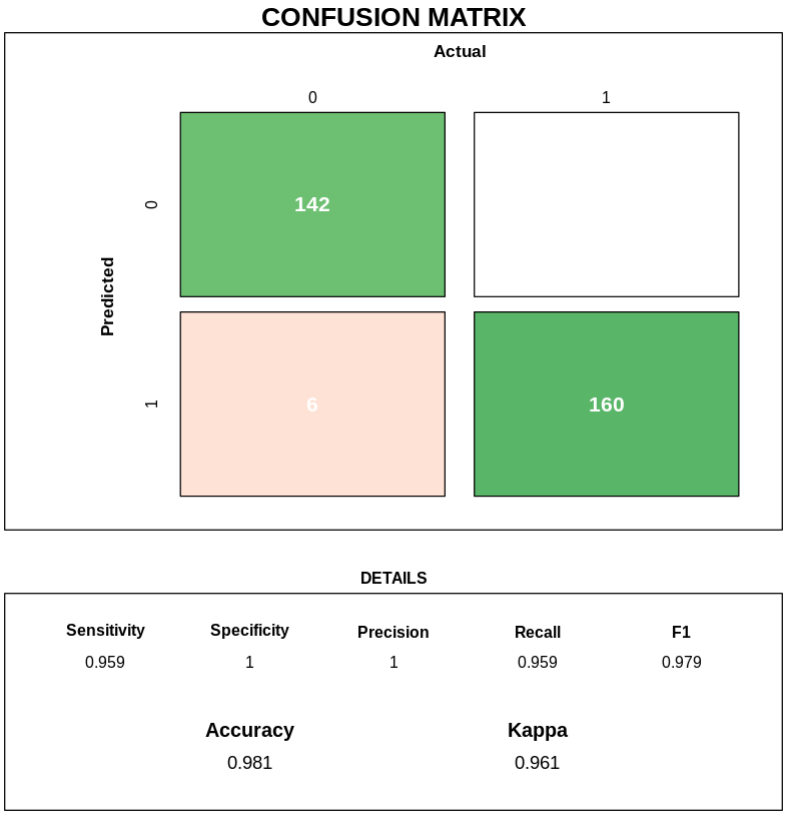
```
randomforestpredict<-predict(bag.heart,heart.test)
randomforestpredict <- ifelse(randomforestpredict > 0.5, 1, 0)
cm<-
confusionMatrix(data=as.factor(randomforestpredict),reference=as.facto
r(heart.test$target1))
```

```
randomforestaccuracy<-mean(randomforestpredict==heart.test$target1)
```

```
randomforestaccuracy
```

```
[1] 0.9805195
```

```
draw_confusion_matrix(cm)
```




```
summary(bag.heart)
```

```
                Length Class  Mode
call              6    -none- call
type              1    -none- character
predicted       717    -none- numeric
mse             100    -none- numeric
rsq             100    -none- numeric
```

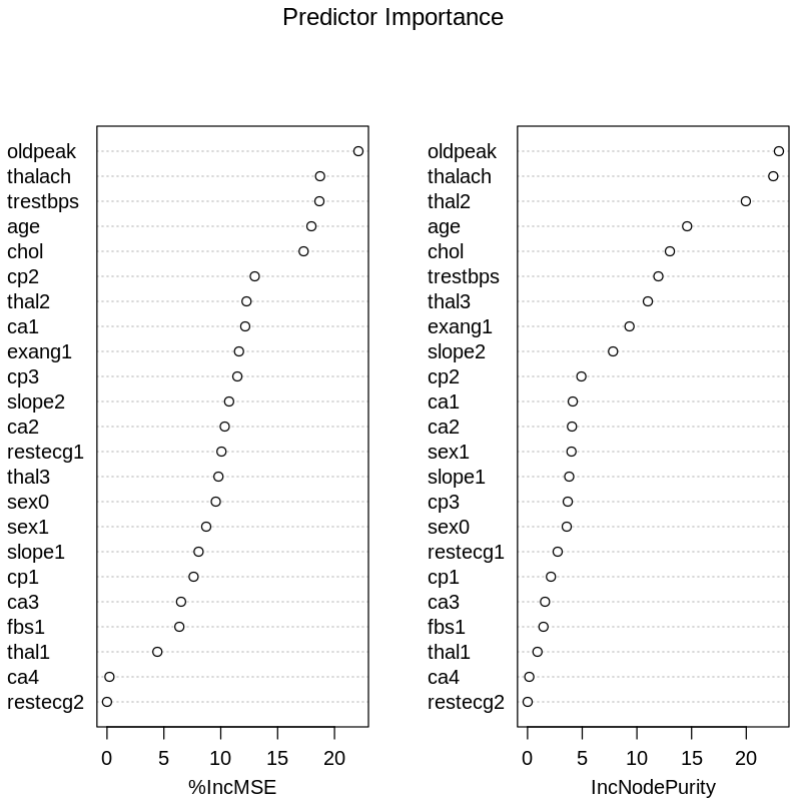
```
oob.times       717    -none- numeric
importance       46    -none- numeric
importanceSD     23    -none- numeric
localImportance   0    -none- NULL
proximity         0    -none- NULL
ntree             1    -none- numeric
mtry              1    -none- numeric
forest           11    -none- list
coefs             0    -none- NULL
y               717    -none- numeric
test              0    -none- NULL
inbag             0    -none- NULL
terms             3    terms  call
```

```
importance<-importance(bag.heart)
```

```
importance
```

```
          %IncMSE    IncNodePurity
age      17.9692447 14.59223130
sex0      9.5635295  3.58466263
sex1      8.7209044  4.01616520
cp1       7.6136227  2.13949528
cp2      12.9890674  4.92208486
cp3      11.4586969  3.67854049
trestbps 18.6731028 11.96179824
chol     17.2859821 13.01639675
fbs1      6.3607371  1.45607112
restecg1 10.0625733  2.75499606
restecg2  0.0000000  0.01004385
thalach  18.7299357 22.47016559
exang1   11.6032109  9.32221798
oldpeak  22.1003386 22.98636683
slope1    8.0518571  3.80983385
slope2   10.7328526  7.81790120
ca1      12.1487972  4.13621118
ca2      10.3503962  4.06272349
ca3       6.5106885  1.59802012
ca4       0.2171856  0.15954128
thal1     4.4287194  0.90696603
thal2    12.2661648 19.96232085
thal3     9.7964155 11.00353702
```

```
varImpPlot(bag.heart,main="Predictor Importance")
```



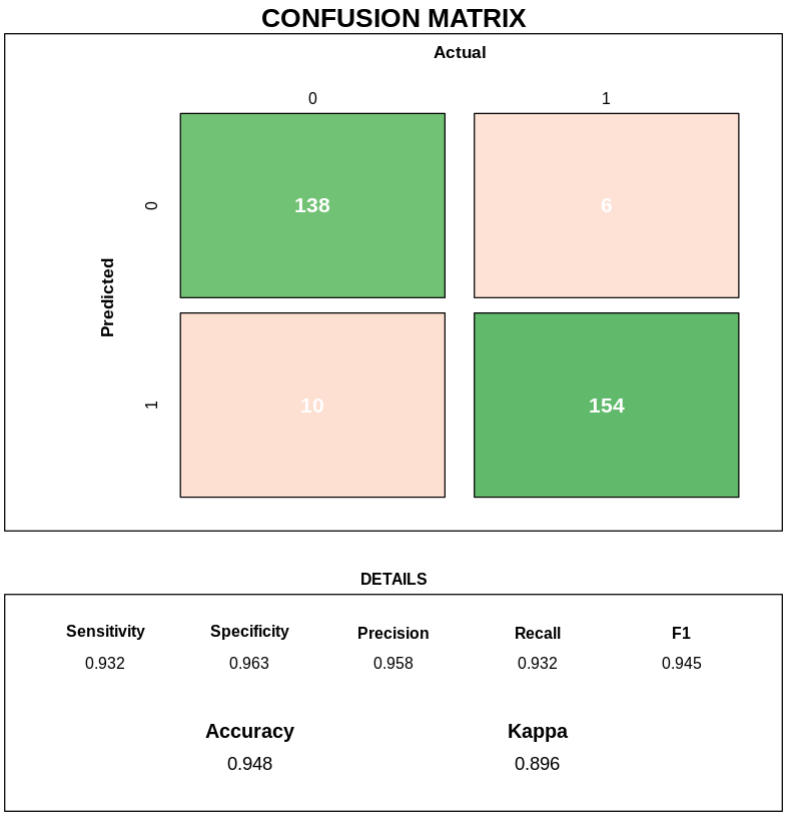
Artificial Neural Network

```
install.packages('neuralnet')
library('neuralnet')
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘Deriv’



Attaching package: ‘neuralnet’


The following object is masked from ‘package:dplyr’:

    compute
```

```
trainlabels <- heart.train$target1
testing_labels <- heart.test$target1
neuralmod = neuralnet(target1~ ., data = heart.train,hidden=6)
neuralpred = predict(neuralmod, newdata = heart.test)
```

```
neuralpred <- ifelse(neuralpred > 0.5, 1, 0)
cm<-
confusionMatrix(data=as.factor(neuralpred),reference=as.factor(testing
_labels))

draw_confusion_matrix(cm)
```




```
neuralnetworkaccuracy<-mean(neuralpred==testing_labels)

neuralnetworkaccuracy

[1] 0.9480519

tribble(
  ~Model,~Accuracy,
  "Logistic Regression", logRegaccuracy,
  "LDA", ldaaccuracy,
  "Naive Bayes", naivebayesaccuracy,
  "Linear SVM", linearsvmaccuracy,
  "Radial SVM", radialsvmaccuracy,
  "KNN", knnaccuracy,
  "Tree", treeaccuracy,
  "ANN", neuralnetworkaccuracy,
)
```

```
  Model               Accuracy
1 Logistic Regression 0.8376623
2 LDA                 0.8538961
3 Naive Bayes         0.8019481
4 Linear SVM          0.8344156
5 Radial SVM          0.8961039
6 KNN                 0.7987013
7 Tree                0.8409091
8 ANN                 0.9480519
```

```
trainlabels <- heart.train$target1
testing_labels <- heart.test$target1
neuralmod = neuralnet(target1~ ., data = heart.train)
neuralpred = predict(neuralmod, newdata = heart.test)
neuralpred <- ifelse(neuralpred > 0.5, 1, 0)
cm<-
confusionMatrix(data=as.factor(neuralpred),reference=as.factor(testing
_labels))

draw_confusion_matrix(cm)
```