

```
# Roll no. 33235
```

```
# Batch: L10
```

```
# PS. Using Linear -Regression and Naive-Bayes algorithm to predict Breast Cancer
```

```
> # Load libraries
```

```
> library('caTools')
```

```
> # Read the dataset
```

```
> breast_cancer = read.csv2 (file.choose(), header = T, sep = ',')
```

```
> names(breast_cancer)
```

```
[1] "x1000025" "x5"      "x1"      "x1.1"    "x1.2"    "x2"      "x1.3"  
    "x3"      "x1.4"    "x1.5"  
[11] "x2.1"
```

```
> # Set the labels
```

```
> names(breast_cancer)[1] = "ID"
```

```
> names(breast_cancer)[2] = "CT" # Clump thickness
```

```
> names(breast_cancer)[3] = "CellSize"
```

```
> names(breast_cancer)[4] = "CellShape"
```

```
> names(breast_cancer)[5] = "MA" # Marginal adhesion
```

```
> names(breast_cancer)[6] = "ECellSize" # Epithelial cell size
```

```
> names(breast_cancer)[7] = "BN" # Bare nuclei
```

```
> names(breast_cancer)[8] = "BC" # Bland chromatin
```

```
> names(breast_cancer)[9] = "NN" # Normal nuclei
```

```
> names(breast_cancer)[10] = "Mit" # Mitoses
```

```
> names(breast_cancer)[11] = "Class" # class
```

```

> names(breast_cancer)
[1] "ID"      "CT"      "CellSize" "CellShape" "MA" "ECellSize" "BN" "BC"
[10] "Mit"     "Class"

> breast_cancer$Class
[1] 22224222222424422424424242222242224244244
[59] 4424424244224244222222222444422222222444
[117] 422224442424222422222222242224224244224222
[175] 424242224424442442222222244222442224424442
[233] 422444422222244222424442222444442444242442
[291] 244242224424244224222422244224224224244422
[349] 422242244444422224422222222224222242222
[407] 2222242424222242224242222224422242222222
[465] 44422222222222422442224442424222222222224
[523] 4222222422222222222222242242222222222222
[581] 4422242424442422222224442242444222222222
[639] 22222222242222222224222222222444222222224
[697] 4 4

> # Set 1 for malignant, 0 for benign (Clean the data)
> breast_cancer$Class <- replace(breast_cancer$Class, breast_cancer$Class == 4,1)
> breast_cancer$Class <- replace(breast_cancer$Class, breast_cancer$Class == 2,0)

> breast_cancer$Class
[1] 000010000001011001011010100000010001011011
[59] 110110101100101100000000011110000000000111
[117] 100001110101000100000000010001001011001000
[175] 101010001101110110000000011000110001101110
[233] 100111100000011000101110000111110111010110
[291] 011010001101011001000100011001001001011100
[349] 10001001111110000110000000000001000010000
[407] 000001010100001000101000000011000100000000
[465] 11100000000000100110001110101000000000001
[523] 10000001000000000000000010010000000000000
[581] 11000101011101000000001110010111000000000
[639] 000000000100000000010000000001110000000001
[697] 1 1

> # Check for missing value
> '?' %in%
breast_cancer$CT [1] FALSE

> '?' %in%
breast_cancer$CellSize [1] FALSE

> '?' %in%
breast_cancer$CellShape [1] FALSE

> '?' %in% breast_cancer$MA
[1] FALSE

> '?' %in%
breast_cancer$ECellSize [1] FALSE

```

```

> '?' %in% breast_cancer$BN # Returned true (16 values are
'?' ) [1] TRUE #i.e. There is a missing value here.

> # replace the NA values
> breast_cancer$BN <- replace(breast_cancer$BN, breast_cancer$BN == '?',NA)
# replace ? with NA
> levels(breast_cancer)[levels(breast_cancer)]
NULL

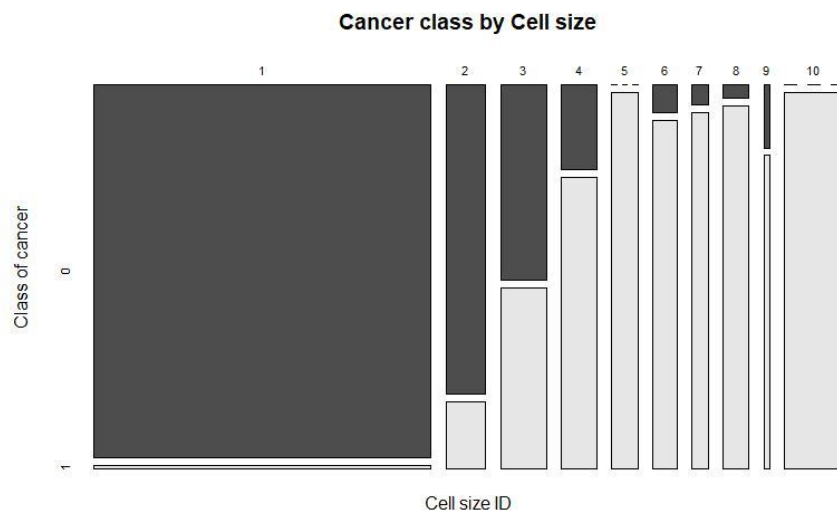
> summary(breast_cancer$CT)
  Min. 1st Qu.  Median Mean 3rd Qu.  Max. 
 2.000  4.000  4.417  6.000 10.000 

> breast_cancer$BN[is.na(breast_cancer$BN)] <- 4.0 # Median value (replace NA)

> # Mosaic plots of some of the factors vs the class of cancer

> mosaicplot(breast_cancer$CellSize ~ breast_cancer$Class, main = "Cancer class by Cell
+ size",color = TRUE, shade = FALSE, xlab = "Cell size ID", ylab = "Class + of
+ cancer")

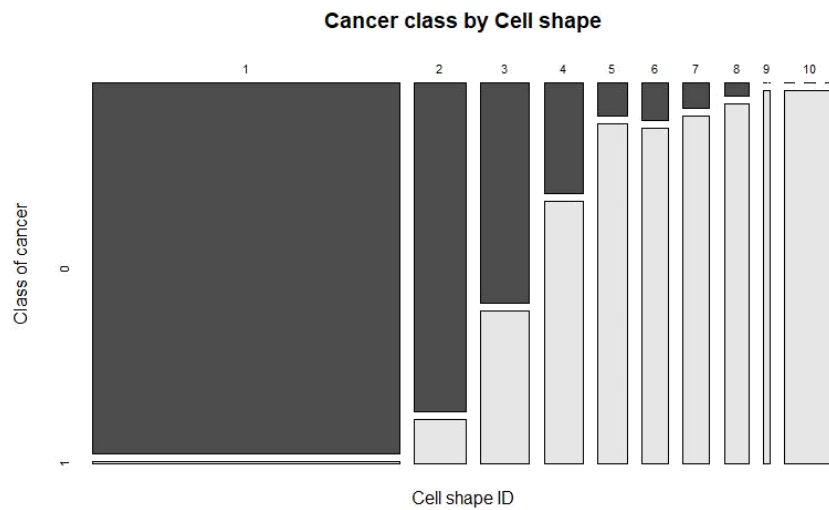
```



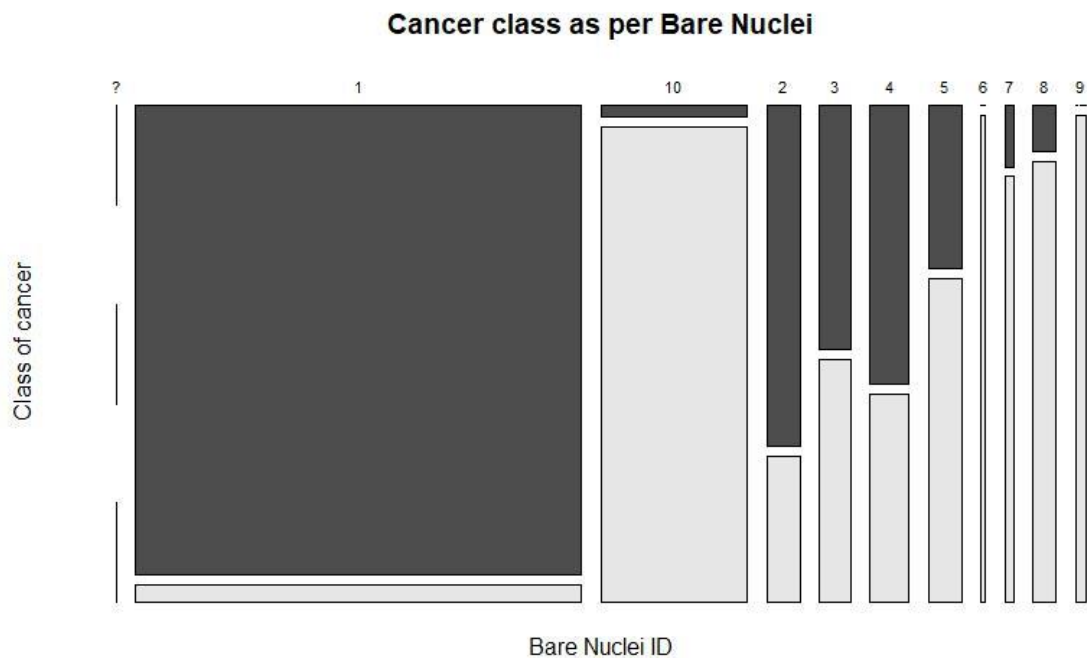
```

> mosaicplot(breast_cancer$CellShape ~ breast_cancer$Class, main = "Cancer class by
+ Cell shape", color = TRUE, shade = FALSE, xlab = "Cell shape ID",
+ ylab = "Class of cancer")

```



```
> mosaicplot(breast_cancer$BN ~ breast_cancer$Class, main = "Cancer class as per
+           Bare Nuclei", color = TRUE, shade = FALSE, xlab = "Bare Nuclei ID",
+           ylab = "Class of cancer")
```



```
> library(caTools) # import library caTools
> set.seed(121)
> # Dividing dataset into training and testing
> split = sample.split(brcdata$Class, SplitRatio = 2/3)
> train_brcdata = subset(brcdata,split == TRUE)
> test_brcdata = subset(brcdata,split == FALSE)
```

```

> > train_brcdata
> ID CT CellSize CellShape MA ECellSize BN BC NN Mit Class
TCellSize
> 1 1002945 5 4 4 5 7 10 3 2 1 0
> 4 3 1016277 6 8 8 1 3 4 3 7 1 0
> 8 5 1017122 8 10 10 8 7 10 9 7 1 1
> 10 6 1018099 1 1 1 1 2 10 3 1 1 0
> 1 7 1018561 2 1 2 1 2 1 3 1 1 0
> 1 8 1033078 2 1 1 1 2 1 1 1 5 0
> 1 9 1033078 4 2 1 1 2 1 2 1 1 0
> 2 10 1035283 1 1 1 1 1 1 3 1 1 0
> 1 11 1036172 2 1 1 1 2 1 2 1 1 0
> 1 14 1044572 8 7 5 10 7 9 5 5 4 1
> 7 17 1049815 4 1 1 1 2 1 3 1 1 0
> 1 18 1050670 10 7 7 6 4 10 4 1 2 1
> 7 19 1050718 6 1 1 1 2 1 3 1 1 0
> 1 20 1054590 7 3 2 10 5 10 5 4 4 1
> 3 22 1056784 3 1 1 1 2 1 2 1 1 0
> 1 23 1057013 8 4 5 1 2 4 7 3 1 1
> 4 24 1059552 1 1 1 1 2 1 3 1 1 0
> 1 25 1065726 5 2 3 4 2 7 3 6 1 1
> 2 26 1066373 3 2 1 1 1 1 2 1 1 0
> 2 27 1066979 5 1 1 1 2 1 2 1 1 0
> 1 31 1071760 2 1 1 1 2 1 3 1 1 0
> 1 32 1072179 10 7 7 3 8 5 7 4 3 1
> 7 33 1074610 2 1 1 2 2 1 3 1 1 0
> 1 34 1075123 3 1 2 1 2 1 2 1 1 0
> 1 35 1079304 2 1 1 1 2 1 2 1 1 0
> 1 37 1081791 6 2 1 1 1 1 7 1 1 0
> 2 38 1084584 5 4 4 9 2 10 5 6 1 1
> 4 39 1091262 2 5 3 3 6 7 7 5 1 1
> 5 40 1096800 6 6 6 9 6 4 7 8 1 0
> 6 41 1099510 10 4 3 1 3 3 6 5 2 1
> 4

```

>6	43	1102573	5	6	5	6	10	1	3	1	1	1
>10	44	1103608	10	10	10	4	8	1	8	10	1	1
>1	45	1103722	1	1	1	1	2	1	2	1	2	0
>7	46	1105257	3	7	7	4	4	9	4	8	1	1
>1	48	1106095	4	1	1	3	2	1	3	1	1	0
>8	49	1106829	7	8	7	2	4	8	3	8	2	1
>5	50	1108370	9	5	8	1	2	3	2	1	5	1
>5	53	1110503	5	5	5	8	10	8	7	3	7	1
>5	54	1110524	10	5	5	6	8	8	7	1	1	1
>6	55	1111249	10	6	6	3	4	5	3	6	1	1
>10	56	1112209	8	10	10	1	3	6	3	9	1	1
>2	57	1113038	8	2	4	1	5	1	5	4	4	1
>2	58	1113483	5	2	3	1	6	10	5	1	1	1
>5	59	1113906	9	5	5	2	2	2	5	1	1	1
>3	60	1115282	5	3	5	5	3	3	4	10	1	1
>10	62	1116116	9	10	10	1	10	8	3	3	1	1
>1	64	1116192	1	1	1	1	2	1	2	1	1	0
>1	66	1117152	4	1	1	1	2	1	3	1	1	0
>3	67	1118039	5	3	4	1	8	10	4	9	1	1
>3	68	1120559	8	3	8	3	4	9	8	9	8	1
>1	69	1121732	1	1	1	1	2	1	3	2	1	0
>10	71	1123061	6	10	2	8	10	2	7	8	10	1
>3	72	1124651	1	3	3	2	2	1	7	2	1	0
>1	75	1131294	1	1	2	1	2	2	4	2	1	0
>1	78	1133136	3	1	1	1	2	3	3	1	1	0
>1	79	1136142	2	1	1	1	3	1	2	1	1	0
>2	80	1137156	2	2	2	1	1	1	7	1	1	0
>1	81	1143978	4	1	1	2	2	1	2	1	1	0
>2	82	1143978	5	2	1	1	2	1	3	1	1	0
>1	83	1147044	3	1	1	1	2	2	7	1	1	0
>5	84	1147699	3	5	7	8	8	9	7	10	7	1
>10	85	1147748	5	10	6	1	10	4	4	10	10	1

```

>      88  1152331  4      1      1  1      2  1  3  1  1  0
>      93  1158247  1      1      1  1      2  1  2  1  1  0
>      94  1160476  2      1      1  1      2  1  3  1  1  0
>      96  1165297  2      1      1  2      2  1  1  1  1  0
>      97  1165790  5      1      1  1      2  1  3  1  1  0
>      98  1165926  9      6      9  2     10  6  2  9 10  1
>      99  1166630  7      5      6 10      5 10  7  9  4  1
>     100  1166654 10      3      5  1     10  5  3 10  2  1
>     102  1167471  4      1      2  1      2  1  3  1  1  0
>     103  1168359  8      2      3  1      6  3  7  1  1  1
>     104  1168736 10     10     10 10     10  1  8  8  8  1
>     106  1170419 10     10     10  8      2 10  4  1  1  1
>     107  1170420  1      6      8 10      8 10  5  7  1  1
>     108  1171710  1      1      1  1      2  1  2  3  1  0
>     109  1171710  6      5      4  4      3  9  7  8  3  1
>     110  1171795  1      3      1  2      2  2  5  3  2  0
>     112  1172152 10      3      3 10      2 10  7  3  3  1
>     113  1173216 10     10     10  3     10  8  8  1  1  1
>     114  1173235  3      3      2  1      2  3  3  1  1  0
>     115  1173347  1      1      1  1      2  5  1  1  1  0
>     117  1173509  4      5      5 10      4 10  7  5  8  1

```

```

>      [ reached 'max' / getOption("max.print") -- omitted 383 rows ]
> regressor=lm(formula = Class~CellShape, data=train_brcdata)
> warning messages:
> 1: In model.response(mf, "numeric") :
>   using type = "numeric" with a factor response will be ignored
> 2: In Ops.factor(y, z$residuals) : '-' not meaningful for factors
> > View(regressor)
> > regressor

```

```

> Call:
> lm(formula = Class ~ CellShape, data = train_brcdata)
>
> Coefficients:
> (Intercept)  CellShape2  CellShape3  CellShape4  CellShape5
CellShape6  CellShape7
> 1.0043      0.1148      0.4124      0.7457      0.8757
0.8905      0.8905
> CellShape8  CellShape9  CellShape10
> 0.9481      0.9957      0.9957
>

```

```

> > brc_shape_predict = predict(regressor, newdata=test_brcdata)
> > brc_shape_predict

```

	2	4	12	13	15	16	21	28
> 29	30							
>	1.004274	1.004274	1.416667	1.004274	1.894737	1.004274	1.880000	1.004274
> 1.416667	1.004274							
>	36	42	47	51	52	61	63	65
> 70	73							
>	2.000000	2.000000	1.004274	1.416667	1.894737	1.004274	1.750000	1.119048
> 1.416667	1.880000							
>	74	76	77	86	87	89	90	91
> 92	95							
>	1.750000	1.750000	1.004274	1.894737	1.894737	1.004274	1.004274	1.004274
> 1.004274	1.004274							
>	101	105	111	116	118	120	124	131
> 133	135							
>	1.750000	1.750000	1.750000	1.416667	1.004274	1.119048	1.894737	1.004274
> 1.004274	1.004274							
>	137	141	147	148	155	156	160	163
> 164	170							
>	1.004274	1.004274	1.004274	1.004274	1.880000	1.119048	1.894737	1.004274
> 1.004274	1.004274							
>	174	175	178	179	181	183	185	188
> 189	191							
>	1.880000	1.894737	1.004274	1.416667	1.004274	1.952381	1.004274	1.750000
> 1.416667	2.000000							
>	195	196	201	203	207	214	218	220
> 221	223							
>	1.004274	1.750000	1.952381	1.004274	1.004274	2.000000	1.894737	1.004274
> 1.750000	1.894737							
>	224	225	228	231	233	235	238	239
> 240	242							
>	1.880000	1.004274	1.004274	1.894737	1.880000	1.750000	2.000000	1.416667
> 1.416667	1.004274							
>	246	249	251	252	257	258	262	263
> 275	276							
>	2.000000	1.004274	1.750000	1.416667	1.004274	1.004274	2.000000	1.750000
> 1.416667	1.004274							
>	277	279	280	291	294	295	296	303
> 306	308							
>	1.004274	1.894737	1.004274	1.004274	1.004274	1.894737	1.750000	1.004274
> 1.004274	1.894737							
>	310	311	318	319	323	324	325	326
> 330	331							
>	1.004274	1.004274	1.004274	1.750000	1.894737	1.004274	1.119048	1.004274
> 1.894737	1.004274							
>	334	338	340	341	342	349	353	361
> 366	370							
>	1.894737	1.004274	1.416667	1.004274	1.004274	1.416667	2.000000	1.416667
> 2.000000	1.119048							
>	373	376	382	384	386	388	389	392
> 405	407							
>	1.004274	1.004274	1.119048	1.004274	1.894737	1.004274	1.004274	1.004274
> 1.004274	1.004274							
>	408	410	412	414	415	417	419	422
> 425	426							
>	1.119048	1.004274	2.000000	1.894737	1.119048	1.004274	1.004274	1.416667
> 2.000000	1.894737							
>	430	438	441	442	445	446	448	456
> 458	459							
>	1.004274	1.416667	1.119048	1.004274	1.004274	1.004274	1.004274	1.880000
> 1.119048	1.416667							
>	464	465	470	473	475	480	484	489
> 491	495							
>	1.004274	1.952381	1.004274	1.004274	1.004274	1.004274	1.119048	1.119048
> 2.000000	1.004274							



```

>          499      502      503      504      505      513      515      519
520      522
>      1.004274 1.004274 1.004274 1.004274 1.004274 1.004274 1.750000 1.952381
1.004274 1.880000
>          529      530      536      540      542      552      553      555
559      560
>      1.004274 1.750000 1.004274 1.004274 1.004274 1.119048 1.119048 1.004274
1.004274 1.004274
>          561      564      565      566      569      575      576      579
582      585
>      1.004274 1.004274 2.000000 1.119048 1.952381 1.119048 1.004274 1.004274
1.880000 1.004274
>          588      594      595      598      599      600      602      605
608      610
>      1.952381 1.894737 1.004274 1.004274 1.750000 1.004274 1.004274 2.000000
2.000000 1.416667
>          612      616      620      621      623      628      630      631
633      636
>      2.000000 1.004274 1.004274 1.416667 1.004274 1.004274 1.416667 1.004274
1.750000 1.894737
>          638      641      654      656      660      666      667      668
669      670
>      1.004274 1.004274 1.004274 1.004274 1.004274 1.119048 1.004274 1.750000
2.000000 1.894737
>          671      674      675      679      680      681      682      683
689      693
>      1.004274 1.004274 1.750000 1.004274 2.000000 2.000000 1.004274 1.004274
1.004274 1.004274
>          695      698
>      1.004274 1.952381
>      > round_shape=brc_shape_predict
>      > r=round(round_shape)
>      > view(r)
>      > r
>      2   4  12  13  15  16  21  28  29  30  36  42  47  51  52  61  63  65
70  73  74  76  77
>      1   1   1   1   2   1   2   1   1   1   2   2   1   1   2   1   2   1
1   2   2   2   1
>      86  87  89  90  91  92  95 101 105 111 116 118 120 124 131 133 135 137
141 147 148 155 156
>      2   2   1   1   1   1   1   2   2   2   1   1   1   2   1   1   1   1
1   1   1   2   1
>      160 163 164 170 174 175 178 179 181 183 185 188 189 191 195 196 201 203
207 214 218 220 221
>      2   1   1   1   2   2   1   1   1   1   2   1   2   1   2   1   2   1
1   2   2   1   2
>      223 224 225 228 231 233 235 238 239 240 242 246 249 251 252 257 258 262
263 275 276 277 279
>      2   2   1   1   2   2   2   2   1   1   1   2   1   2   1   1   1   2
2   1   1   1   2
>      280 291 294 295 296 303 306 308 310 311 318 319 323 324 325 326 330 331
334 338 340 341 342
>      1   1   1   2   2   1   1   2   1   1   1   2   2   1   1   1   2   1
2   1   1   1   1
>      349 353 361 366 370 373 376 382 384 386 388 389 392 405 407 408 410 412
414 415 417 419 422
>      1   2   1   2   1   1   1   1   1   2   1   1   1   1   1   1   1   2
2   1   1   1   1
>      425 426 430 438 441 442 445 446 448 456 458 459 464 465 470 473 475 480
484 489 491 495 499
>      2   2   1   1   1   1   1   1   1   2   1   1   1   2   1   1   1   1
1   1   2   1   1
>      502 503 504 505 513 515 519 520 522 529 530 536 540 542 552 553 555 559
560 561 564 565 566

```

```

> 1 1 1 2 1 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1
> 569 575 576 579 582 585 588 594 595 598 599 600 602 605 608 610 612 616
620 621 623 628 630
> 2 1 1 2 1 2 2 1 1 2 1 1 2 2 1 2 1
> 1 1 1 1 1
> 631 633 636 638 641 654 656 660 666 667 668 669 670 671 674 675 679 680
681 682 683 689 693
> 1 2 2 1 1 1 1 1 1 1 2 2 2 1 1 2 1 2
> 2 1 1 1
> 695 698
> 1 2
> > table(r,test_brcdata$class)
>
> r 0 1
> 1 144 11
> 2 8 69
> > library(e1071)
> library(caret)
> typeof(r)
> [1] "double"
> levels(r)
> NULL
> levels(test_brcdata$class)
> [1] "0" "1"
> str(r)
> Named num [1:232] 1 1 1 1 2 1 2 1 1 1 12 13 ...
> - attr(*, "names")= chr [1:232] "2" "4" "12" "13" ...
> r2 = as.data.frame(r)
> r2
> r
> 2 1
> 4 1
> 12 1
> 13 1
> 15 2
> 16 1
> 21 2
> 28 1
> 29 1
> 30 1
> 36 2
> 42 2
> 47 1
> 51 1
> 52 2
> 61 1
> 63 2
> 65 1
> 70 1
> 73 2
> 74 2
> 76 2
> 77 1
> 86 2
> 87 2
> 89 1
> 90 1
> 91 1
> 92 1
> 95 1
> 101 2
> 105 2
> 111 2

```

>	116	1
>	118	1
>	120	1
>	124	2
>	131	1
>	133	1
>	135	1
>	137	1
>	141	1
>	147	1
>	148	1
>	155	2
>	156	1
>	160	2
>	163	1
>	164	1
>	170	1
>	174	2
>	175	2
>	178	1
>	179	1
>	181	1
>	183	2
>	185	1
>	188	2
>	189	1
>	191	2
>	195	1
>	196	2
>	201	2
>	203	1
>	207	1
>	214	2
>	218	2
>	220	1
>	221	2
>	223	2
>	224	2
>	225	1
>	228	1
>	231	2
>	233	2
>	235	2
>	238	2
>	239	1
>	240	1
>	242	1
>	246	2
>	249	1
>	251	2
>	252	1
>	257	1
>	258	1
>	262	2
>	263	2
>	275	1
>	276	1
>	277	1
>	279	2
>	280	1
>	291	1
>	294	1
>	295	2
>	296	2

>	303	1
>	306	1
>	308	2
>	310	1
>	311	1
>	318	1
>	319	2
>	323	2
>	324	1
>	325	1
>	326	1
>	330	2
>	331	1
>	334	2
>	338	1
>	340	1
>	341	1
>	342	1
>	349	1
>	353	2
>	361	1
>	366	2
>	370	1
>	373	1
>	376	1
>	382	1
>	384	1
>	386	2
>	388	1
>	389	1
>	392	1
>	405	1
>	407	1
>	408	1
>	410	1
>	412	2
>	414	2
>	415	1
>	417	1
>	419	1
>	422	1
>	425	2
>	426	2
>	430	1
>	438	1
>	441	1
>	442	1
>	445	1
>	446	1
>	448	1
>	456	2
>	458	1
>	459	1
>	464	1
>	465	2
>	470	1
>	473	1
>	475	1
>	480	1
>	484	1
>	489	1
>	491	2
>	495	1
>	499	1

>	502	1
>	503	1
>	504	1
>	505	1
>	513	1
>	515	2
>	519	2
>	520	1
>	522	2
>	529	1
>	530	2
>	536	1
>	540	1
>	542	1
>	552	1
>	553	1
>	555	1
>	559	1
>	560	1
>	561	1
>	564	1
>	565	2
>	566	1
>	569	2
>	575	1
>	576	1
>	579	1
>	582	2
>	585	1
>	588	2
>	594	2
>	595	1
>	598	1
>	599	2
>	600	1
>	602	1
>	605	2
>	608	2
>	610	1
>	612	2
>	616	1
>	620	1
>	621	1
>	623	1
>	628	1
>	630	1
>	631	1
>	633	2
>	636	2
>	638	1
>	641	1
>	654	1
>	656	1
>	660	1
>	666	1
>	667	1
>	668	2
>	669	2
>	670	2
>	671	1
>	674	1
>	675	2
>	679	1
>	680	2

```

> 681 2
> 682 1
> 683 1
> 689 1
> 693 1
> 695 1
> 698 2
> > df2=confusionMatrix(as.factor(r2$r),as.factor(test_brcdata$class))
> Error in confusionMatrix.default(as.factor(r2$r),
as.factor(test_brcdata$class)) :
>   The data contain levels not found in the data.
> > df2
> Confusion Matrix and Statistics
>
>           Reference
> Prediction    0    1
>           0 151   24
>           1   1   56
>
>           Accuracy : 0.8922
>           95% CI   : (0.845, 0.929)
>           No Information Rate : 0.6552
>           P-Value [Acc > NIR] : < 2.2e-16
>
>           Kappa : 0.7441
>
>           Mcnemar's Test P-Value : 1.083e-05
>
>           Sensitivity : 0.9934
>           Specificity : 0.7000
>           Pos Pred Value : 0.8629
>           Neg Pred Value : 0.9825
>           Prevalence : 0.6552
>           Detection Rate : 0.6509
>           Detection Prevalence : 0.7543
>           Balanced Accuracy : 0.8467
>
>           'Positive' Class : 0
>
> # Create the dataframes for training and testing
> brcdata<-breast_cancer
> brcdata$ID=factor(brcdata$ID)
> brcdata$CT=factor(brcdata$CT)

```

```

> brcdata$TCellSize=factor(brcdata$CellSize)
> brcdata$CellShape=factor(brcdata$CellShape)
> brcdata$MA=factor(brcdata$MA)
> brcdata$ECellSize=factor(brcdata$ECellSize)
> brcdata$BN=factor(brcdata$BN)
> brcdata$BC=factor(brcdata$BC)
> brcdata$NN=factor(brcdata$NN)
> brcdata$Mit=factor(brcdata$Mit)
> brcdata$Class=factor(brcdata$Class)

> # Dividing dataset into training and testing
> split = sample.split(brcdata$Class, SplitRatio = 2/3)
> train_brcdata = subset(brcdata,split == TRUE)
> test_brcdata = subset(brcdata,split == FALSE)

```

```

> train_brcdata

```

	ID	CT	cellSize	CellShape	MA	ECellSize	BN	BC	NN	Mit	Class	TCellSize
1	1002945	5	4	4	5	7	10	3	2	1	0	4
2	1015425	3	1	1	1	2	2	3	1	1	0	1
8	1033078	2	1	1	1	2	1	1	1	5	0	1
9	1033078	4	2	1	1	2	1	2	1	1	0	2
10	1035283	1	1	1	1	1	1	3	1	1	0	1
11	1036172	2	1	1	1	2	1	2	1	1	0	1
12	1041801	5	3	3	3	2	3	4	4	1	1	3
14	1044572	8	7	5	10	7	9	5	5	4	1	7
15	1047630	7	4	6	4	6	1	4	3	1	1	4
16	1048672	4	1	1	1	2	1	2	1	1	0	1
18	1050670	10	7	7	6	4	10	4	1	2	1	7
19	1050718	6	1	1	1	2	1	3	1	1	0	1
21	1054593	10	5	5	3	6	7	7	10	1	1	5
22	1056784	3	1	1	1	2	1	2	1	1	0	1
23	1057013	8	4	5	1	2	4	7	3	1	1	4
24	1059552	1	1	1	1	2	1	3	1	1	0	1
25	1065726	5	2	3	4	2	7	3	6	1	1	2
27	1066979	5	1	1	1	2	1	2	1	1	0	1
28	1067444	2	1	1	1	2	1	2	1	1	0	1
29	1070935	1	1	3	1	2	1	1	1	1	0	1
30	1070935	3	1	1	1	1	1	2	1	1	0	1
31	1071760	2	1	1	1	2	1	3	1	1	0	1
33	1074610	2	1	1	2	2	1	3	1	1	0	1
35	1079304	2	1	1	1	2	1	2	1	1	0	1
36	1080185	10	10	10	8	6	1	8	9	1	1	10
37	1081791	6	2	1	1	1	1	7	1	1	0	2
38	1084584	5	4	4	9	2	10	5	6	1	1	4
39	1091262	2	5	3	3	6	7	7	5	1	1	5
40	1096800	6	6	6	9	6	4	7	8	1	0	6
41	1099510	10	4	3	1	3	3	6	5	2	1	4
42	1100524	6	10	10	2	8	10	7	3	3	1	10
45	1103722	1	1	1	1	2	1	2	1	2	0	1
47	1105524	1	1	1	1	2	1	2	1	1	0	1
49	1106829	7	8	7	2	4	8	3	8	2	1	8
51	1108449	5	3	3	4	2	4	3	4	1	1	3
52	1110102	10	3	6	2	3	5	4	10	2	1	3
53	1110503	5	5	5	8	10	8	7	3	7	1	5
54	1110524	10	5	5	6	8	8	7	1	1	1	5

56	1112209	8	10	10	1	3	6	3	9	1	1	10
57	1113038	8	2	4	1	5	1	5	4	4	1	2
58	1113483	5	2	3	1	6	10	5	1	1	1	2
59	1113906	9	5	5	2	2	2	5	1	1	1	5
60	1115282	5	3	5	5	3	3	4	10	1	1	3
61	1115293	1	1	1	1	2	2	2	1	1	0	1
63	1116132	6	3	4	1	5	2	3	9	1	1	3
64	1116192	1	1	1	1	2	1	2	1	1	0	1
65	1116998	10	4	2	1	3	2	4	3	10	1	4
66	1117152	4	1	1	1	2	1	3	1	1	0	1
67	1118039	5	3	4	1	8	10	4	9	1	1	3
68	1120559	8	3	8	3	4	9	8	9	8	1	3
70	1121919	5	1	3	1	2	1	2	1	1	0	1
71	1123061	6	10	2	8	10	2	7	8	10	1	10
75	1131294	1	1	2	1	2	2	4	2	1	0	1
76	1132347	1	1	4	1	2	1	2	1	1	0	1
77	1133041	5	3	1	2	2	1	2	1	1	0	3
78	1133136	3	1	1	1	2	3	3	1	1	0	1
79	1136142	2	1	1	1	3	1	2	1	1	0	1
80	1137156	2	2	2	1	1	1	7	1	1	0	2
81	1143978	4	1	1	2	2	1	2	1	1	0	1
84	1147699	3	5	7	8	8	9	7	10	7	1	5
85	1147748	5	10	6	1	10	4	4	10	10	1	10
88	1152331	4	1	1	1	2	1	3	1	1	0	1
90	1156272	1	1	1	1	2	1	3	1	1	0	1
91	1156948	3	1	1	2	2	1	1	1	1	0	1
92	1157734	4	1	1	1	2	1	3	1	1	0	1
96	1165297	2	1	1	2	2	1	1	1	1	0	1
97	1165790	5	1	1	1	2	1	3	1	1	0	1
98	1165926	9	6	9	2	10	6	2	9	10	1	6
99	1166630	7	5	6	10	5	10	7	9	4	1	5
101	1167439	2	3	4	4	2	5	2	5	1	1	3
103	1168359	8	2	3	1	6	3	7	1	1	1	2
104	1168736	10	10	10	10	10	1	8	8	8	1	10
105	1169049	7	3	4	4	3	3	3	2	7	1	3
106	1170419	10	10	10	8	2	10	4	1	1	1	10
107	1170420	1	6	8	10	8	10	5	7	1	1	6
110	1171795	1	3	1	2	2	2	5	3	2	0	3
111	1171845	8	6	4	3	5	9	3	1	1	1	6
112	1172152	10	3	3	10	2	10	7	3	3	1	3
114	1173235	3	3	2	1	2	3	3	1	1	0	3
115	1173347	1	1	1	1	2	5	1	1	1	0	1
117	1173509	4	5	5	10	4	10	7	5	8	1	5
119	1173681	3	2	1	1	2	2	3	1	1	0	2
120	1174057	1	1	2	2	2	1	3	1	1	0	1

[ reached 'max' / getOption("max.print") -- omitted 383 rows ]



```
> # Applying Naive Bayes classifier on dataset
> library(e1071)
> classifier <- naiveBayes(Class ~ CT+CellSize+CellShape+MA+ECellSize+BN+BC+NN+Mit,
                           train_brcdata)
> classifier
```

Naive Bayes Classifier for Discrete Predictors

Call:  
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

```
Y
      0      1
0.6545064 0.3454936
```

Conditional probabilities:

```
CT
Y      1      2      3      4      5      6      7
      8      9
0 0.318032787 0.101639344 0.196721311 0.157377049 0.173770492 0.042622951 0.003278689
0.006557377 0.000000000
1 0.018633540 0.012422360 0.031055901 0.049689441 0.198757764 0.074534161 0.105590062
0.167701863 0.043478261
```

```
CT
Y      10
0 0.000000000
1 0.298136646
```

```
cellSize
Y      [,1]      [,2]
0 1.334426 0.9352183
1 6.515528 2.7502400
```

```
CellShape
Y      1      2      3      4      5      6      7
      8      9
0 0.777049180 0.118032787 0.055737705 0.032786885 0.003278689 0.006557377 0.006557377
0.000000000 0.000000000
1 0.006211180 0.031055901 0.118012422 0.118012422 0.136645963 0.093167702 0.099378882
0.136645963 0.031055901
```

```
cellShape
Y      10
0 0.000000000
1 0.229813665
```

```
MA
Y      1      2      3      4      5      6      7
      8      9
0 0.819672131 0.081967213 0.062295082 0.013114754 0.013114754 0.006557377 0.000000000
0.000000000 0.003278689
1 0.136645963 0.099378882 0.080745342 0.111801242 0.093167702 0.055900621 0.062111801
0.118012422 0.012422360
```

```
MA
Y      10
0 0.000000000
1 0.229813665
```

0 0.000000000  
1 0.229813665

ECellSize  
Y 1 2 3 4 5 6 7  
8 9  
0 0.104918033 0.780327869 0.072131148 0.009836066 0.013114754 0.006557377 0.006557377  
0.003278689 0.000000000  
1 0.006211180 0.111801242 0.167701863 0.167701863 0.149068323 0.173913043 0.018633540  
0.080745342 0.012422360

ECellSize  
Y 10  
0 0.003278689  
1 0.111801242

BN  
Y 1 10 2 3 4 5 6  
7 8  
0 0.852459016 0.006557377 0.039344262 0.026229508 0.039344262 0.026229508 0.000000000  
0.003278689 0.006557377  
1 0.068322981 0.540372671 0.055900621 0.037267081 0.062111801 0.074534161 0.018633540  
0.037267081 0.062111801

BN  
Y 9  
0 0.000000000  
1 0.043478261

BC  
Y 1 2 3 4 5 6 7  
8 9  
0 0.337704918 0.344262295 0.262295082 0.022950820 0.013114754 0.003278689 0.016393443  
0.000000000 0.000000000  
1 0.012422360 0.037267081 0.130434783 0.136645963 0.130434783 0.018633540 0.310559006  
0.111801242 0.043478261

BC  
Y 10  
0 0.000000000  
1 0.068322981

NN  
Y 1 2 3 4 5 6 7  
8 9  
0 0.865573770 0.075409836 0.029508197 0.003278689 0.003278689 0.003278689 0.003278689  
0.013114754 0.003278689  
1 0.173913043 0.018633540 0.167701863 0.062111801 0.080745342 0.055900621 0.062111801  
0.062111801 0.074534161

NN  
Y 10  
0 0.000000000  
1 0.242236025

Mit  
Y 1 2 3 4 5 6 7  
8 10  
0 0.957377049 0.026229508 0.006557377 0.000000000 0.003278689 0.000000000 0.003278689

```
0.003278689 0.000000000
1 0.552795031 0.111801242 0.130434783 0.037267081 0.012422360 0.012422360 0.024844720
0.037267081 0.080745342
```

```
> #predict using trained model
> prediction <- predict(classifier, test_brcdata ,type="class")
```

```
> table(prediction, test_brcdata[,11]) # put it in table
```

```
prediction    0    1
           0 147    0
           1   5   80
```

```
> # Displaying the accuracy using confusion Matrix
> library(e1071)
> library(caret)
```

```
> df1=confusionMatrix(test_brcdata[,11],prediction) #create the confusion matrix
```

```
> df1
```

```
Confusion Matrix and Statistics
```

```
          Reference
Prediction    0    1
           0 147    5
           1   0   80
```

```
          Accuracy : 0.9784
          95% CI   : (0.9504, 0.993)
No Information Rate : 0.6336
P-Value [Acc > NIR] : < 2e-16
```

```
          Kappa : 0.953
```

```
McNemar's Test P-Value : 0.07364
```

```
          Sensitivity : 1.0000
          Specificity : 0.9412
          Pos Pred Value : 0.9671
          Neg Pred Value : 1.0000
          Prevalence : 0.6336
          Detection Rate : 0.6336
          Detection Prevalence : 0.6552
          Balanced Accuracy : 0.9706
```

```
'Positive' Class : 0
```

```
>
```