Project Report of capstone

Humaira Rizwan

November 13, 2018

Defining Problem

To predict that patient of PIMA Indians Diabetes Dataset has diabetes or not based on different attributes/information within the database. And to explore whether there is relation between weight and diabetes.

Data

This dataset is from National Institude od Digestive and Kidney Disease.All patients is this data set are specifically women and minimum of age 21 years older of Pima heritage.

Loading Data of Pima Medical Dataset.

```
mydata<-read.csv("C:/Users/ADMIN/downloads/diabetes.csv",header=TRUE,sep=",")
```

Exploratory Analysis.

Check the dimensionality.

```
dim(mydata)
```

```
## [1] 768   9
```

Structure

```
str(mydata)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

Based on structure we observe that Pima Indians Diabetes Database has 768 obs and 9 varibles.Outcome is 0 and 1 , so it is binary classification problem.And data is quanitative except outcome it should be qualitative.

Attributes of Pima Indians Diabetes Database.

```
attributes(mydata)
```

```
## $names
## [1] "Pregnancies"              "Glucose"
## [3] "BloodPressure"            "SkinThickness"
## [5] "Insulin"                  "BMI"
## [7] "DiabetesPedigreeFunction" "Age"
## [9] "Outcome"
##
## $class
## [1] "data.frame"
##
## $row.names
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
##  [35]  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  [52]  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
##  [69]  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##  [86]  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
## [103] 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
## [120] 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
## [137] 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
## [154] 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170
## [171] 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
## [188] 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204
## [205] 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221
## [222] 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
## [239] 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
## [256] 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272
## [273] 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289
## [290] 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
## [307] 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
## [324] 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
## [341] 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357
## [358] 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374
## [375] 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391
## [392] 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408
## [409] 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
## [426] 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442
## [443] 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459
## [460] 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476
## [477] 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493
## [494] 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510
## [511] 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527
## [528] 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544
## [545] 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561
## [562] 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578
## [579] 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595
## [596] 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612
## [613] 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629
## [630] 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646
## [647] 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663
```

```
## [664] 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680
## [681] 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697
## [698] 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714
## [715] 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731
## [732] 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748
## [749] 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765
## [766] 766 767 768
```

Head of data

```
head(mydata)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

Tail of data

```
tail(mydata)
```

```
##     Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 763           9      89            62             0       0 22.5
## 764          10     101            76            48     180 32.9
## 765           2     122            70            27       0 36.8
## 766           5     121            72            23     112 26.2
## 767           1     126            60             0       0 30.1
## 768           1      93            70            31       0 30.4
##     DiabetesPedigreeFunction Age Outcome
## 763                    0.142  33       0
## 764                    0.171  63       0
## 765                    0.340  27       0
## 766                    0.245  30       0
## 767                    0.349  47       1
## 768                    0.315  23       0
```

The variables represents:

Pregnant - number times pregnancy occur

Glucose _ plasma glucose concentration at 2 hours in an oral glucose tolerance test

BloodPressure _ diastolic blood pressure (mmHg)

Skin Thickness _ triceps is skin fold thickness (mm)

Insulin _ 2-hour serum insulin (mu U/ml)

BMI _ the body mass index (weight in kg/(height in m2))

Diabetes pedigree function *pedigree Dieabetes #####Age* define age (years).

Outcome _ is the result as they are diabetic or not.

test whether the patient showed signs of diabetes (0=negative, 1=positive)

summary.

```
summary(mydata)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

As we see min and max values of summary we find that Glucose , BloodPressure , #####SkinThickness , Insulin ,BMI have 0 values.So pregnacy can be 0 for woman who is not

pregnent.

Explore more to find missing values.

```
sort(mydata$Insulin)
```

```
##   [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [35]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [52]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [69]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [86]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [103]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [120]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [137]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [154]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [171]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [188]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [205]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [222]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [239]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [256]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [273]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [290]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [307]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [324]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [341]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [358]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [375]  14  15  16  18  18  22  23  23  25  29  32  36  36  36  37  37  38
## [392]  40  40  41  42  43  44  44  44  45  45  45  46  48  48  48  49  49
## [409]  49  49  49  50  50  50  51  52  53  53  54  54  54  54  55  55  56
## [426]  56  56  56  56  57  57  58  58  59  60  60  61  63  63  63  64  64
## [443]  64  64  65  66  66  66  66  66  67  67  68  70  70  70  71  71  71
## [460]  71  72  73  74  74  74  75  75  75  76  76  76  76  76  77  77  78
## [477]  78  79  79  81  82  82  82  83  83  83  84  85  85  86  87  87  88
## [494]  88  88  88  89  90  90  90  90  91  92  92  92  94  94  94  94  94
## [511]  94  94  95  95  96  96  99  99 100 100 100 100 100 100 100 105 105
## [528] 105 105 105 105 105 105 105 105 105 106 106 106 108 110 110 110 110
## [545] 110 110 112 114 114 115 115 115 115 115 115 116 116 119 120 120 120
## [562] 120 120 120 120 120 122 122 125 125 125 125 126 126 126 127 128 129
## [579] 130 130 130 130 130 130 130 130 130 132 132 135 135 135 135 135 135
## [596] 140 140 140 140 140 140 140 140 140 142 144 144 145 145 145 146 148
## [613] 148 150 150 152 152 155 155 155 155 156 156 156 158 158 159 160 160
## [630] 160 160 165 165 165 165 166 167 167 168 168 168 168 170 170 171 175
## [647] 175 175 176 176 176 178 180 180 180 180 180 180 180 182 182 182 183
## [664] 184 185 185 188 190 190 190 190 191 192 192 193 194 194 194 196 200
## [681] 200 200 200 204 205 205 207 207 210 210 210 210 210 215 215 215 220
## [698] 220 225 225 228 230 230 231 231 235 237 240 240 245 249 250 255 258
## [715] 265 265 270 271 272 274 275 277 278 280 284 285 285 291 293 293 300
## [732] 304 310 318 321 325 325 325 326 328 330 335 342 360 370 375 387 392
## [749] 402 415 440 465 474 478 480 480 485 495 495 510 540 543 545 579 600
## [766] 680 744 846
```

```
sort(mydata$BloodPressure)
```

```
##   [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [35]   0  24  30  30  38  40  44  44  44  44  46  46  48  48  48  48  48
##  [52]  50  50  50  50  50  50  50  50  50  50  50  50  50  52  52  52  52
##  [69]  52  52  52  52  52  52  52  54  54  54  54  54  54  54  54  54  54
##  [86]  54  55  55  56  56  56  56  56  56  56  56  56  56  56  56  56  58  58
## [103]  58  58  58  58  58  58  58  58  58  58  58  58  58  58  58  58  58
## [120]  58  58  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
## [137]  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60  60
## [154]  60  60  60  60  60  61  62  62  62  62  62  62  62  62  62  62  62  62
## [171]  62  62  62  62  62  62  62  62  62  62  62  62  62  62  62  62  62  62
## [188]  62  62  62  62  62  62  64  64  64  64  64  64  64  64  64  64  64  64
## [205]  64  64  64  64  64  64  64  64  64  64  64  64  64  64  64  64  64  64
## [222]  64  64  64  64  64  64  64  64  64  64  64  64  64  64  64  65  65
## [239]  65  65  65  65  65  66  66  66  66  66  66  66  66  66  66  66  66  66
## [256]  66  66  66  66  66  66  66  66  66  66  66  66  66  66  66  66  66  66
## [273]  66  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68
## [290]  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68  68
## [307]  68  68  68  68  68  68  68  68  68  68  68  68  68  70  70  70  70  70
## [324]  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
## [341]  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
## [358]  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70  70
## [375]  70  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72
## [392]  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72  72
## [409]  72  72  72  72  72  72  72  72  72  72  72  72  74  74  74  74  74  74
## [426]  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74
## [443]  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74  74
## [460]  74  74  74  74  74  74  74  74  74  74  74  74  74  75  75  75  75  75
## [477]  75  75  75  76  76  76  76  76  76  76  76  76  76  76  76  76  76  76
## [494]  76  76  76  76  76  76  76  76  76  76  76  76  76  76  76  76  76  76
## [511]  76  76  76  76  76  76  76  76  78  78  78  78  78  78  78  78  78  78
## [528]  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78
## [545]  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78  78
## [562]  78  78  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80
## [579]  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80
## [596]  80  80  80  80  80  80  80  80  82  82  82  82  82  82  82  82  82  82
## [613]  82  82  82  82  82  82  82  82  82  82  82  82  82  82  82  82  82  82
## [630]  82  82  82  82  84  84  84  84  84  84  84  84  84  84  84  84  84  84
## [647]  84  84  84  84  84  84  84  84  84  84  85  85  85  85  85  85  86
## [664]  86  86  86  86  86  86  86  86  86  86  86  86  86  86  86  86  86  86
## [681]  86  86  86  88  88  88  88  88  88  88  88  88  88  88  88  88  88  88
## [698]  88  88  88  88  88  88  88  88  88  88  88  90  90  90  90  90  90
## [715]  90  90  90  90  90  90  90  90  90  90  90  90  90  90  90  90  90  92
## [732]  92  92  92  92  92  92  92  94  94  94  94  94  94  95  96  96  96
## [749]  96  98  98  98 100 100 100 102 104 104 106 106 106 108 108 110 110
## [766] 110 114 122
```

Based upon our results it is found that zero has been used as a missing value code so we change 0 in to NULL.

Modeling data

BloodPressure

```
mydata$BloodPressure[mydata$BloodPressure == 0] <- NA
```

Glucose

```
mydata$Glucose[mydata$Glucose== 0] <- NA
```

SkinThickness

```
mydata$SkinThickness[mydata$SkinThickness==0] <- NA
```

Insulin

```
mydata$Insulin[mydata$Insulin==0] <- NA
```

BMI

```
mydata$BMI[mydata$BMI==0] <- NA
```

Chang the name of outcome into diabetic so can understand easily result.

```
colnames(mydata)[9] <- "diabetic"
head(mydata)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35      NA 33.6
## 2           1      85            66            29      NA 26.6
## 3           8     183            64            NA      NA 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74            NA      NA 25.6
##   DiabetesPedigreeFunction Age diabetic
## 1                    0.627  50        1
## 2                    0.351  31        0
## 3                    0.672  32        1
## 4                    0.167  21        0
## 5                    2.288  33        1
## 6                    0.201  30        0
```

Now change 0 and 1 value from diabetic into 1=yes and 0=no for better understanding.

```
mydata$diabetic <- as.factor(mydata$diabetic)

levels(mydata$diabetic) <- c("N0","Yes")
tail(mydata)
```

```
##     Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 763           9      89            62            NA      NA 22.5
## 764          10     101            76            48     180 32.9
## 765           2     122            70            27      NA 36.8
## 766           5     121            72            23     112 26.2
## 767           1     126            60            NA      NA 30.1
## 768           1      93            70            31      NA 30.4
##     DiabetesPedigreeFunction Age diabetic
## 763                    0.142  33       N0
## 764                    0.171  63       N0
## 765                    0.340  27       N0
## 766                    0.245  30       N0
## 767                    0.349  47      Yes
## 768                    0.315  23       N0
```

Now again check summary of data .

```
summary(mydata)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   :29.15
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##                   NA's   :5       NA's   :35       NA's   :227
##     Insulin            BMI        DiabetesPedigreeFunction      Age
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0780           Min.   :21.00
##  1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437           1st Qu.:24.00
##  Median :125.00   Median :32.30   Median :0.3725           Median :29.00
##  Mean   :155.55   Mean   :32.46   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##  NA's   :374      NA's   :11
##  diabetic
##  N0 :500
##  Yes:268
##
##
##
##
##
```

Now 0 value in minimum of Glucose , Bloodpressure , SkinThickness Insulin ,Bmi , has been change .

Find Correlation Between Variable

-

```
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 3.4.4
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

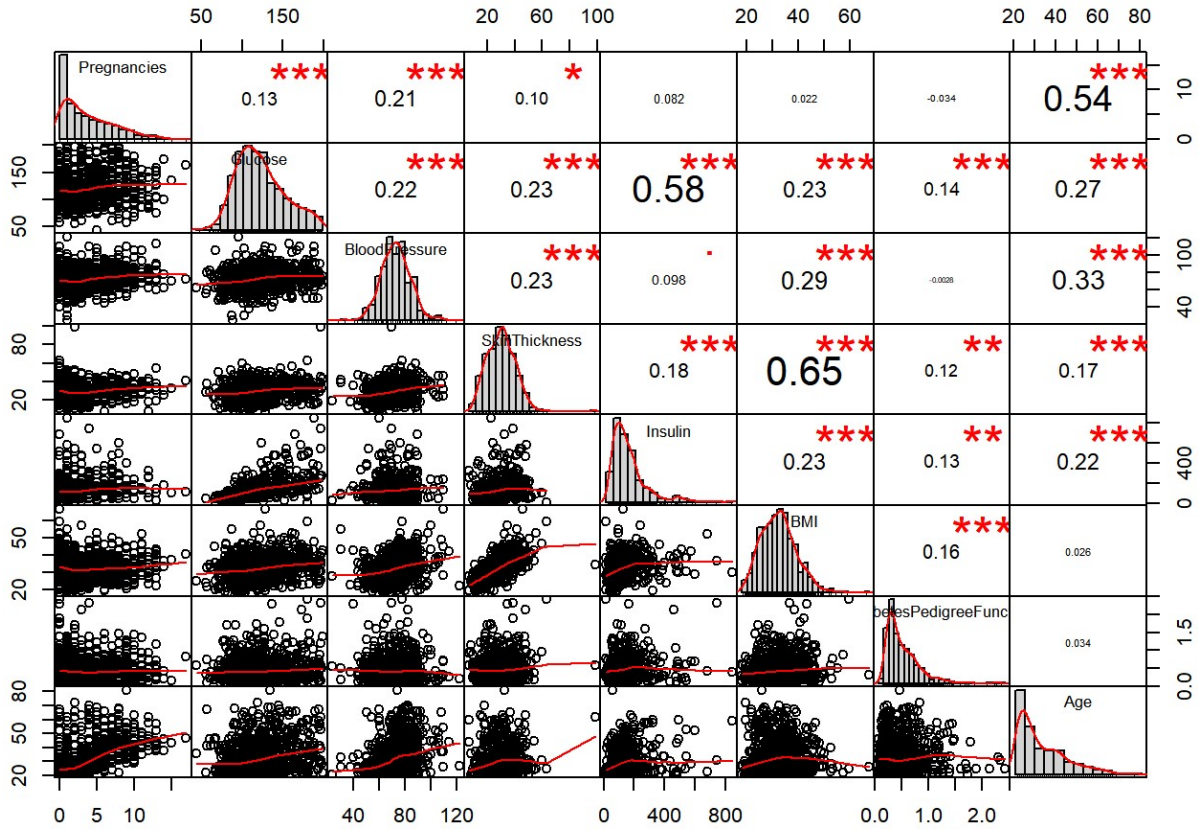
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
chart.Correlation(mydata[,-9], histogram=TRUE, col="grey10", pch=1, main="Chart Correlati
on of Variance")
```



Explore the relationship between variables

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
install.packages('ggplot2', dependencies=TRUE, repos='http://cran.rstudio.com/')
```

```
## Installing package into 'C:/Users/ADMIN/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\ADMIN\AppData\Local\Temp\RtmpQ1NeAp\downloaded_packages
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

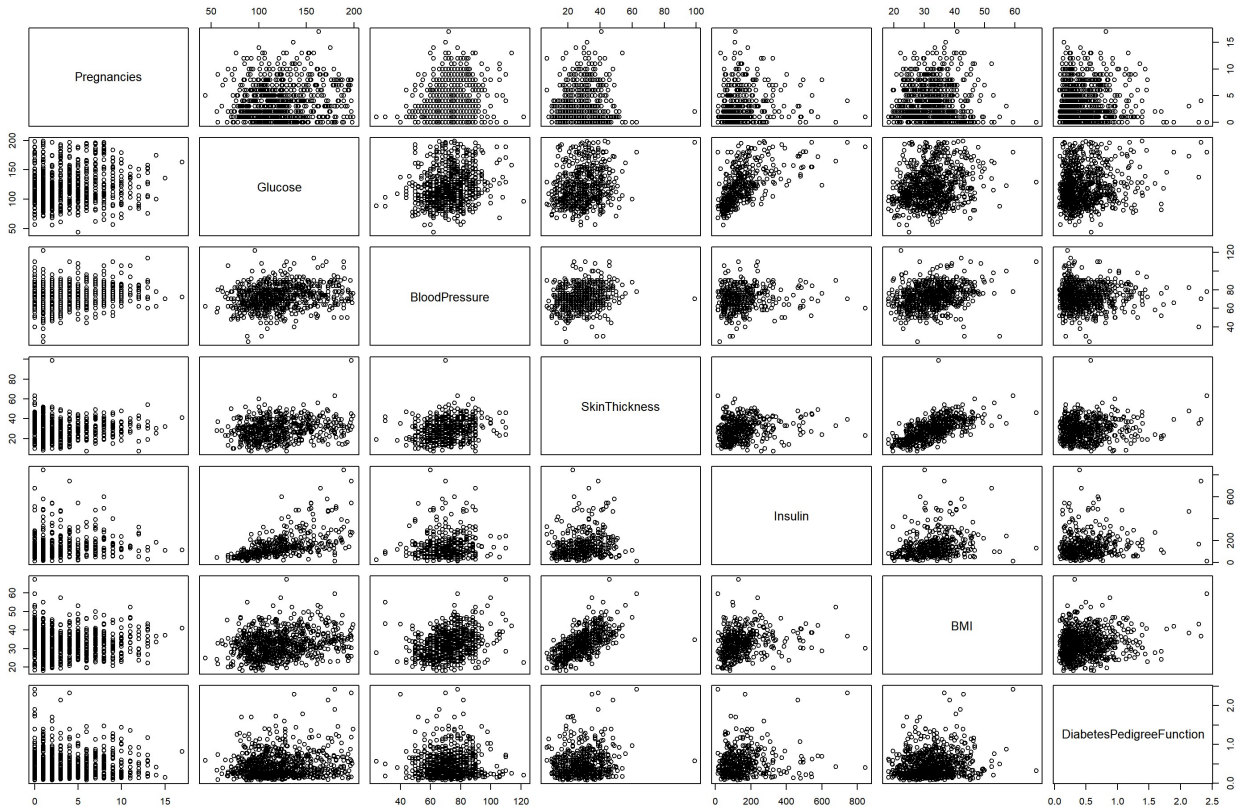
```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.4.4
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
pairs(mydata [,1:7])
plot(mydata [,1:7])
```




```
library(ggplot2)

library(GGally)

ggpairs(mydata, aes(color=diabetic, alpha=0.75), lower=list(continuous="smooth"))+ theme_
bw()+

labs(title="Correlation Plot of Variance(diabetes)")+

theme(plot.title=element_text(face='bold',color='black',hjust=0.5,size=12))
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 35 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 227 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 374 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 11 rows containing missing values
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 40 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 232 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 375 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 16 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 35 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```

```
## Warning: Removed 40 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 40 rows containing missing values (geom_point).
```

```
## Warning: Removed 35 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 229 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 374 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 39 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 35 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 35 rows containing missing values
```

```
## Warning: Removed 35 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 227 rows containing missing values (geom_point).
```

```
## Warning: Removed 232 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 232 rows containing missing values (geom_point).
```

```
## Warning: Removed 229 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 229 rows containing missing values (geom_point).
```

```
## Warning: Removed 227 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 374 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 229 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 227 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 227 rows containing missing values
```

```
## Warning: Removed 227 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 374 rows containing missing values (geom_point).
```

```
## Warning: Removed 375 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 375 rows containing missing values (geom_point).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 374 rows containing missing values (geom_point).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 374 rows containing missing values (geom_point).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 375 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 374 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 374 rows containing missing values
```

```
## Warning: Removed 374 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## Warning: Removed 16 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Warning: Removed 39 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 39 rows containing missing values (geom_point).
```

```
## Warning: Removed 229 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 229 rows containing missing values (geom_point).
```

```
## Warning: Removed 375 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 375 rows containing missing values (geom_point).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_density).
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 11 rows containing missing values
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 11 rows containing missing values
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 35 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```

```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 227 rows containing missing values (geom_point).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 374 rows containing missing values (geom_point).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 35 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```

```
## Warning: Removed 227 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 227 rows containing missing values (geom_point).
```

```
## Warning: Removed 374 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 374 rows containing missing values (geom_point).
```

```
## Warning: Removed 11 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 35 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 227 rows containing non-finite values (stat_bin).
```

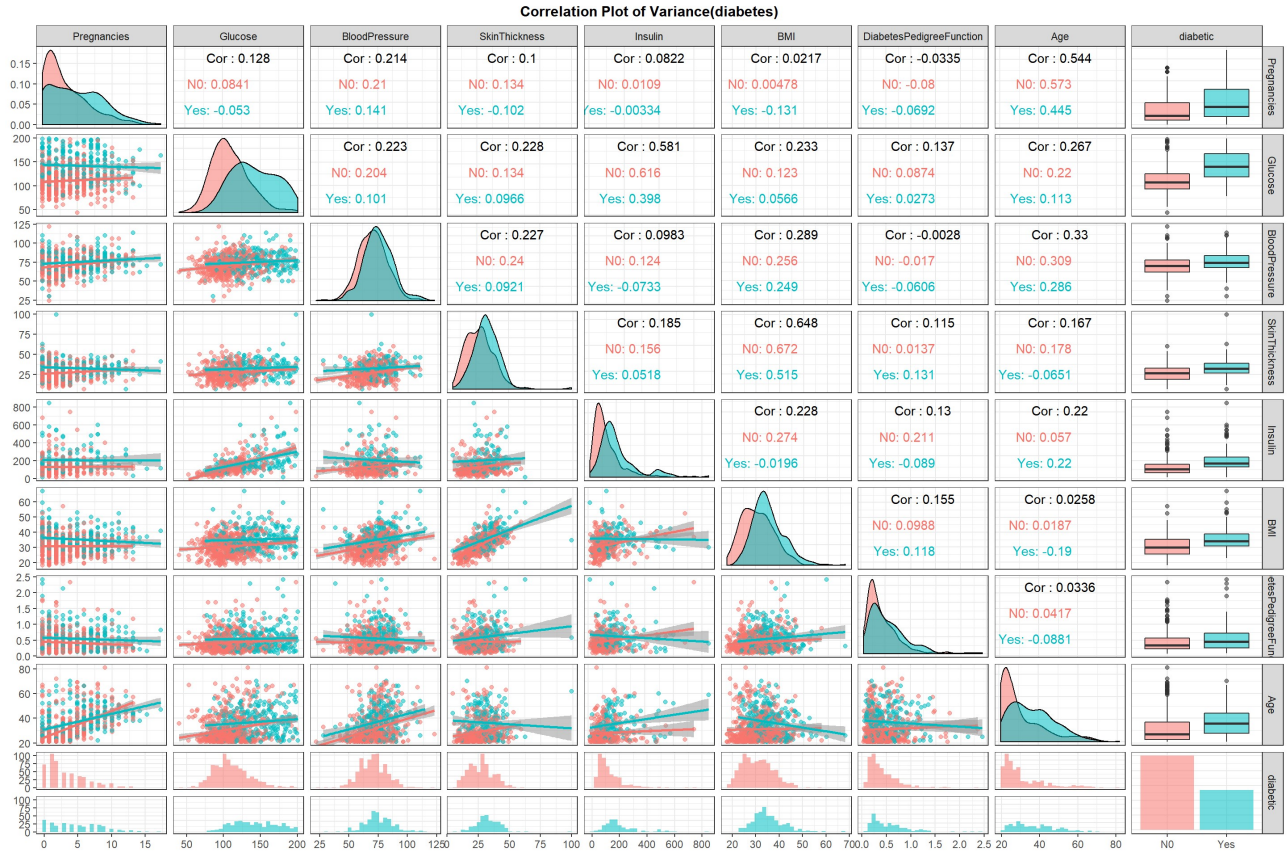
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 374 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 11 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



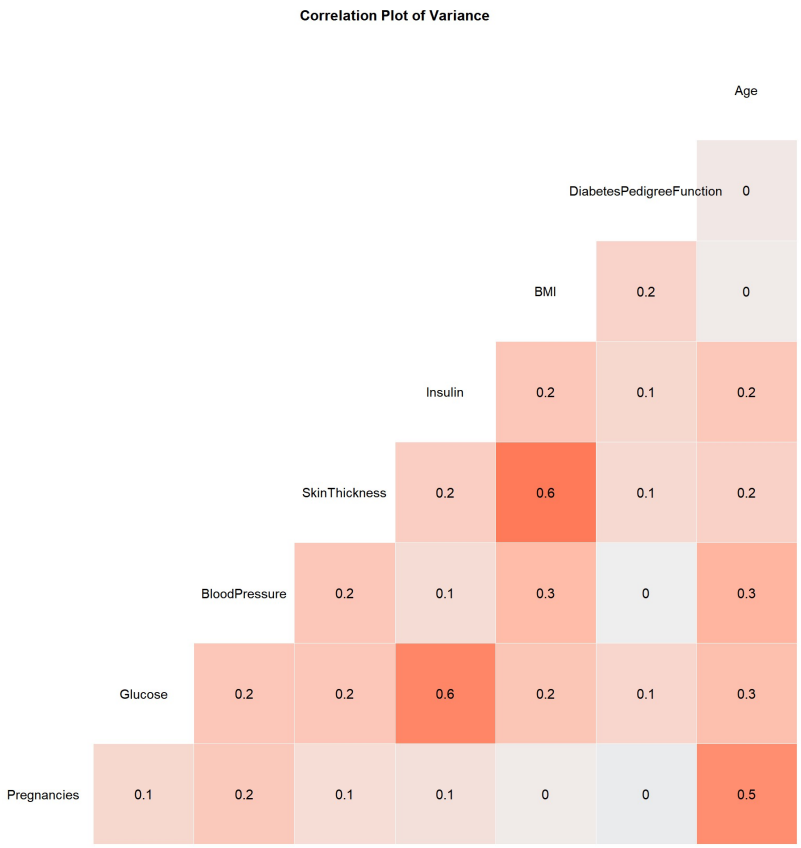
####By ggcorr, we can see high correlation in below variance

```
library(GGally)
ggcorr(mydata[,-9], name = "corr", label = TRUE)+

  theme(legend.position="none")+

labs(title="Correlation Plot of Variance")+

theme(plot.title=element_text(face='bold',color='black',hjust=0.5,size=12))
```



pregnencies and Age are 0.5 correlated => About 50% correlated to each other.

Skin thickness , BMI and Insulin 0.4 About 40% correlated to each other.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

```
correlationMatrix <- cor(mydata[,1:8])
print(correlationMatrix)
```

```
##                           Pregnancies Glucose BloodPressure SkinThickness
## Pregnancies                1.00000000      NA             NA            NA
## Glucose                           NA       1             NA            NA
## BloodPressure                     NA      NA              1            NA
## SkinThickness                     NA      NA             NA             1
## Insulin                           NA      NA             NA            NA
## BMI                               NA      NA             NA            NA
## DiabetesPedigreeFunction -0.03352267      NA             NA            NA
## Age                       0.54434123      NA             NA            NA
##                           Insulin BMI DiabetesPedigreeFunction        Age
## Pregnancies                    NA  NA               -0.03352267 0.54434123
## Glucose                        NA  NA                        NA         NA
## BloodPressure                  NA  NA                        NA         NA
## SkinThickness                  NA  NA                        NA         NA
## Insulin                         1  NA                        NA         NA
## BMI                            NA   1                        NA         NA
## DiabetesPedigreeFunction       NA  NA                1.00000000 0.03356131
## Age                            NA  NA                0.03356131 1.00000000
```

Calculate number of diebetic patient with result yes or No

Make test & train dataset

```
nrows <- NROW(mydata)

set.seed(218)                           # fix random value

index <- sample(1:nrows, 0.7 * nrows)   # shuffle and divide



# train <- diab                         # 768 test data (100%)

train <- mydata[index,]                   # 537 test data (70%)

test <- mydata[-index,]                   # 231 test data (30%)
```

TRAIN

```
prop.table(table(train$diabetic))
```

```
##
##        N0       Yes
## 0.6648045 0.3351955
```

TEST

```
prop.table(table(test$diabetic))
```

```
##
##        N0       Yes
## 0.6190476 0.3809524
```

Different Machine Learning Methods for Binary Classification Problem

1-Recursive partiting for classification

```
library(rpart)

explore_rp <- rpart(diabetic~.,data=train,control=rpart.control(minsplit=2))

pre_rp <- predict(explore_rp, test[,-9], type="class")

confusionmatrix_rp  <- confusionMatrix(pre_rp, test$diabetic)

confusionmatrix_rp
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N0 Yes
##        N0  119  39
##        Yes  24  49
##
##                Accuracy : 0.7273
##                  95% CI : (0.665, 0.7836)
##     No Information Rate : 0.619
##     P-Value [Acc > NIR] : 0.0003497
##
##                   Kappa : 0.4022
##  Mcnemar's Test P-Value : 0.0777599
##
##             Sensitivity : 0.8322
##             Specificity : 0.5568
##          Pos Pred Value : 0.7532
##          Neg Pred Value : 0.6712
##              Prevalence : 0.6190
##          Detection Rate : 0.5152
##    Detection Prevalence : 0.6840
##       Balanced Accuracy : 0.6945
##
##        'Positive' Class : N0
##
```

2-Prune classification

```
explore_pru <- prune(explore_rp,cp=explore_rp$cptable[which.min(explore_rp$cptable[,"xerr
or"]),"CP"])

pre_prune <- predict(explore_pru, test[,-9], type="class")

confusionmatrix_pru <-confusionMatrix(pre_prune, test$diabetic)

confusionmatrix_pru
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N0 Yes
##        N0  130  56
##        Yes  13  32
##
##                Accuracy : 0.7013
##                  95% CI : (0.6378, 0.7596)
##     No Information Rate : 0.619
##     P-Value [Acc > NIR] : 0.005522
##
##                   Kappa : 0.301
##  Mcnemar's Test P-Value : 4.277e-07
##
##             Sensitivity : 0.9091
##             Specificity : 0.3636
##          Pos Pred Value : 0.6989
##          Neg Pred Value : 0.7111
##              Prevalence : 0.6190
##          Detection Rate : 0.5628
##    Detection Prevalence : 0.8052
##       Balanced Accuracy : 0.6364
##
##        'Positive' Class : N0
##
```

3-Naive_Bays Algorithum

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.4.4
```

```
##
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:PerformanceAnalytics':
##
##     kurtosis, skewness
```

```
library(caret)
explore_nb <- naiveBayes(train[,-9], train$diabetic)

prep_nb <- predict(explore_nb, test[,-9])

confusionmatrix_nb <- confusionMatrix(prep_nb, test$diabetic)

confusionmatrix_nb
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N0 Yes
##        N0  114  38
##        Yes  29  50
##
##                Accuracy : 0.71
##                  95% CI : (0.6468, 0.7676)
##     No Information Rate : 0.619
##     P-Value [Acc > NIR] : 0.002381
##
##                   Kappa : 0.3727
##  Mcnemar's Test P-Value : 0.328393
##
##             Sensitivity : 0.7972
##             Specificity : 0.5682
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.6329
##              Prevalence : 0.6190
##          Detection Rate : 0.4935
##    Detection Prevalence : 0.6580
##       Balanced Accuracy : 0.6827
##
##        'Positive' Class : N0
##
```

4_C5.0 Algothrium

```
library(C50)
```

```
## Warning: package 'C50' was built under R version 3.4.4
```

```
mydata$diabetic
```

```
##   [1] Yes N0  Yes N0  Yes N0  Yes N0  Yes Yes N0  Yes N0  Yes Yes Yes Yes
##  [18] Yes N0  Yes N0  N0  Yes Yes Yes Yes Yes N0  N0  N0  N0  Yes N0  N0
##  [35] N0  N0  N0  Yes Yes Yes N0  N0  N0  Yes N0  Yes N0  N0  Yes N0  N0
##  [52] N0  N0  Yes N0  N0  Yes N0  N0  N0  N0  Yes N0  N0  Yes N0  Yes N0
##  [69] N0  N0  Yes N0  Yes N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  Yes
##  [86] N0  N0  N0  Yes N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  Yes Yes N0
## [103] N0  N0  N0  N0  N0  N0  N0  Yes Yes Yes N0  N0  Yes Yes Yes N0  N0
## [120] N0  Yes N0  N0  N0  Yes Yes N0  N0  Yes Yes Yes Yes Yes N0  N0  N0
## [137] N0  N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  N0  N0  N0  Yes
## [154] N0  Yes Yes N0  N0  N0  Yes N0  N0  N0  N0  Yes Yes N0  N0  N0  N0
## [171] Yes Yes N0  N0  N0  Yes N0  Yes N0  Yes N0  N0  N0  N0  N0  Yes Yes
## [188] Yes Yes Yes N0  N0  Yes Yes N0  Yes N0  Yes Yes Yes N0  N0  N0  N0
## [205] N0  N0  Yes Yes N0  Yes N0  N0  N0  Yes Yes Yes Yes N0  Yes Yes Yes
## [222] Yes N0  N0  N0  N0  N0  Yes N0  N0  Yes Yes N0  N0  N0  Yes Yes Yes
## [239] Yes N0  N0  N0  Yes Yes N0  Yes N0  N0  N0  N0  N0  N0  N0  N0  Yes
## [256] Yes N0  N0  N0  Yes N0  Yes N0  N0  Yes N0  Yes N0  N0  Yes Yes N0
## [273] N0  N0  N0  N0  Yes N0  N0  N0  Yes N0  N0  Yes Yes N0  N0  Yes N0
## [290] N0  N0  Yes Yes Yes N0  N0  Yes N0  Yes N0  Yes Yes N0  Yes N0  N0
## [307] Yes N0  Yes Yes N0  N0  Yes N0  Yes N0  N0  Yes N0  Yes N0  Yes Yes
## [324] Yes N0  N0  Yes N0  Yes N0  N0  N0  Yes N0  N0  N0  N0  Yes Yes Yes
## [341] N0  N0  N0  N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  Yes Yes
## [358] Yes N0  Yes Yes N0  N0  Yes N0  N0  Yes N0  N0  Yes Yes N0  N0  N0
## [375] N0  Yes N0  N0  Yes N0  N0  N0  N0  N0  N0  N0  Yes Yes Yes N0  N0
## [392] Yes N0  N0  Yes N0  N0  Yes N0  Yes Yes N0  Yes N0  Yes N0  Yes N0
## [409] Yes Yes N0  N0  N0  N0  Yes Yes N0  Yes N0  Yes N0  N0  N0  N0  Yes
## [426] Yes N0  Yes N0  Yes N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  Yes N0
## [443] N0  Yes Yes Yes N0  N0  Yes N0  N0  Yes N0  N0  N0  Yes N0  N0  Yes
## [460] N0  N0  N0  N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  N0  N0
## [477] Yes N0  N0  N0  Yes N0  N0  N0  Yes Yes N0  N0  N0  N0  N0  N0  N0
## [494] Yes N0  N0  N0  N0  Yes N0  N0  N0  Yes N0  N0  N0  Yes N0  N0  N0
## [511] Yes N0  N0  N0  N0  Yes Yes N0  N0  N0  N0  N0  N0  Yes N0  N0  N0
## [528] N0  N0  N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  Yes Yes Yes Yes N0
## [545] N0  Yes Yes N0  N0  N0  N0  N0  N0  N0  N0  N0  N0  N0  N0  N0  Yes
## [562] Yes N0  N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  N0  N0  N0  N0  Yes
## [579] N0  Yes Yes N0  N0  N0  Yes N0  Yes N0  Yes N0  Yes N0  Yes N0  N0
## [596] Yes N0  N0  Yes N0  N0  N0  N0  Yes Yes N0  Yes N0  N0  N0  N0  Yes
## [613] Yes N0  Yes N0  N0  N0  Yes Yes N0  N0  N0  N0  N0  N0  N0  N0  N0
## [630] N0  Yes N0  N0  N0  N0  Yes N0  N0  Yes N0  N0  N0  Yes N0  N0  N0
## [647] Yes Yes Yes N0  N0  N0  N0  N0  N0  Yes N0  N0  N0  Yes N0  Yes Yes
## [664] Yes Yes N0  Yes Yes N0  N0  N0  N0  N0  N0  N0  Yes Yes N0  Yes N0
## [681] N0  Yes N0  Yes N0  N0  N0  N0  N0  Yes N0  Yes N0  Yes N0  Yes Yes
## [698] N0  N0  N0  N0  Yes Yes N0  N0  N0  Yes N0  Yes Yes N0  N0  Yes N0
## [715] N0  Yes Yes N0  N0  Yes N0  N0  Yes N0  N0  N0  N0  N0  N0  N0  Yes
## [732] Yes Yes N0  N0  N0  N0  N0  N0  Yes Yes N0  N0  Yes N0  N0  Yes N0
## [749] Yes Yes Yes N0  N0  Yes Yes Yes N0  Yes N0  Yes N0  Yes N0  N0  N0
## [766] N0  Yes N0
## Levels: N0 Yes
```

```
acc_test <- numeric()

accuracy1 <- NULL; accuracy2 <- NULL



for(i in 1:50){

    learn_imp_c50 <- C5.0(train[,-9],train$diabetic,trials = i)

    p_c50 <- predict(learn_imp_c50, test[,-9])

    accuracy1 <- confusionMatrix(p_c50, test$diabetic)

    accuracy2[i] <- accuracy1$overall[1]

}



acc <- data.frame(t= seq(1,50), cnt = accuracy2)



opt_t <- subset(acc, cnt==max(cnt))[1,]

sub <- paste("Optimal number of trials is", opt_t$t, "(accuracy :", opt_t$cnt,") in C5.
0")




learn_imp_c50 <- C5.0(train[,-9],train$diabetic,trials=opt_t$t)

pre_imp_c50 <- predict(learn_imp_c50, test[,-9])

cm_imp_c50 <- confusionMatrix(pre_imp_c50, test$diabetic)

cm_imp_c50
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N0 Yes
##        N0  124  34
##        Yes  19  54
##
##                Accuracy : 0.7706
##                  95% CI : (0.7109, 0.8232)
##     No Information Rate : 0.619
##     P-Value [Acc > NIR] : 6.567e-07
##
##                   Kappa : 0.4971
##  Mcnemar's Test P-Value : 0.05447
##
##             Sensitivity : 0.8671
##             Specificity : 0.6136
##          Pos Pred Value : 0.7848
##          Neg Pred Value : 0.7397
##              Prevalence : 0.6190
##          Detection Rate : 0.5368
##    Detection Prevalence : 0.6840
##       Balanced Accuracy : 0.7404
##
##        'Positive' Class : N0
##
```

5_ADA Boost Classifier

```
library(rpart)

library(ada)
```

```
## Warning: package 'ada' was built under R version 3.4.4
```

```
control <- rpart.control(cp = -1, maxdepth = 14,maxcompete = 1,xval = 0)

explore_ada <- ada(diabetic~., data = train, test.x = train[,-9], test.y = train[,9], typ
e = "gentle", control = control, iter = 70)

pre_ada <- predict(explore_ada, test[,-9])

confusionmatrix_ada <- confusionMatrix(pre_ada, test$diabetic)

confusionmatrix_ada
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  N0 Yes
##        N0  122  39
##        Yes  21  49
##
##                Accuracy : 0.7403
##                  95% CI : (0.6787, 0.7956)
##     No Information Rate : 0.619
##     P-Value [Acc > NIR] : 6.69e-05
##
##                   Kappa : 0.4268
##  Mcnemar's Test P-Value : 0.02819
##
##             Sensitivity : 0.8531
##             Specificity : 0.5568
##          Pos Pred Value : 0.7578
##          Neg Pred Value : 0.7000
##              Prevalence : 0.6190
##          Detection Rate : 0.5281
##    Detection Prevalence : 0.6970
##       Balanced Accuracy : 0.7050
##
##        'Positive' Class : N0
##
```

Choose Best Machine Learning Algorithum and their summary

Visulization for comparing Accuracy

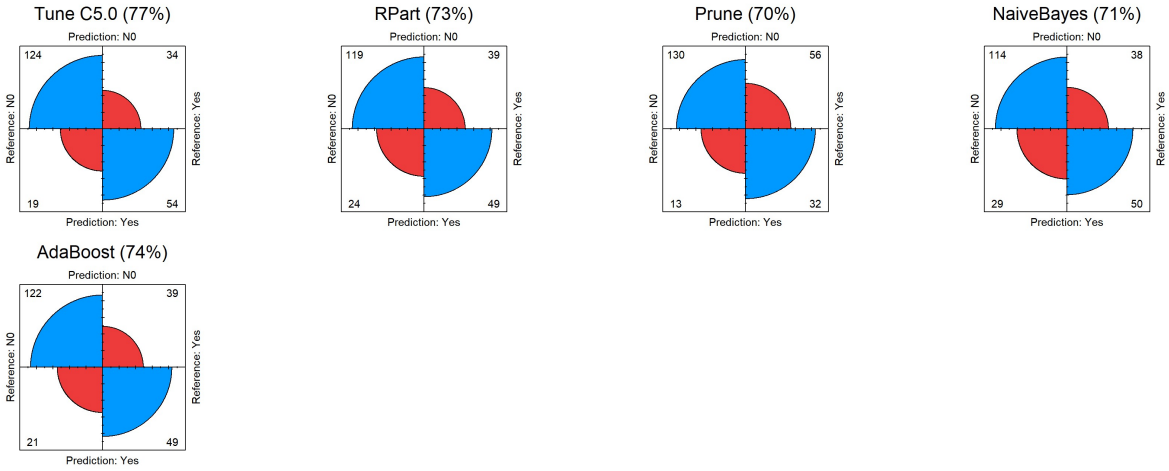
```
col <- c("#ed3b3b", "#0099ff")

par(mfrow=c(3,4))

fourfoldplot(cm_imp_c50$table, color = col, conf.level = 0, margin = 1, main=paste("Tune
C5.0 (",round(cm_imp_c50$overall[1]*100),"%)",sep=""))

fourfoldplot(confusionmatrix_rp$table, color = col, conf.level = 0, margin = 1, main=past
e("RPart (",round(confusionmatrix_rp$overall[1]*100),"%)",sep=""))

fourfoldplot(confusionmatrix_pru$table, color = col, conf.level = 0, margin = 1, main=pas
te("Prune (",round(confusionmatrix_pru$overall[1]*100),"%)",sep=""))
fourfoldplot(confusionmatrix_nb$table, color = col, conf.level = 0, margin = 1, main=past
e("NaiveBayes (",round(confusionmatrix_nb$overall[1]*100),"%)",sep=""))
fourfoldplot(confusionmatrix_ada$table, color = col, conf.level = 0, margin = 1, main=pas
te("AdaBoost (",round(confusionmatrix_ada$overall[1]*100),"%)",sep=""))
```



Choose best Model

-

```
opt_predict <- c( cm_imp_c50$overall[1],confusionmatrix_rp$overall[1], confusionmatrix_pr
u$overall[1],  confusionmatrix_nb$overall[1],  confusionmatrix_ada$overall[1])

names(opt_predict) <- c("C50","rpart","prune","nb","ada")

best_predict_model <- subset(opt_predict, opt_predict==max(opt_predict))

best_predict_model
```

```
##       C50
## 0.7705628
```

Test our Model prediction

-

Choosing one patient at one time help to diagnosis patient is he diabetic .

check Patient data for testing function

YES Diabetic patients

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.4.4
```

```
Y <- test[1,]                  ## 5th patient

kable(Y )
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	diabetic
5	0	137	40	35	16843.1		2.288	33	Yes

NO Diabetic patients

```
N <- test[2,]                  ## 18th patient

kable(N)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	diabetic
6	5	116	74	NA	NA	25.6	0.201	30	N0

For purpose of testing remove Diebetic coloumn

```
Y$diabetic <- NULL

N$diabetic <- NULL
```

Predicting Model

- Make a function for predicting diabetic patient.

```
diabetic_patient_predict <- function(new, method=explore_ada) {

    new_pre <- predict(method, new)

    new_res <- as.character(new_pre)

    return(paste("Result: ", new_res, sep=""))

}
```

Testing function

```
diabetic_patient_predict(Y)
```

```
## [1] "Result: Yes"
```

Use ada boost to predict

```
diabetic_patient_predict(Y,explore_ada)
```

```
## [1] "Result: Yes"
```

Predicting test Diabetic.

Use C5.0

```
pre <- data.frame(orgin_result = test$diabetic, predict_result = pre_imp_c50 , correct =
ifelse(test$diabetic == pre_imp_c50 , "True", "False"))

kable(head(sub,10))
```

x

Optimal number of trials is 9 (accuracy : 0.770562770562771) in C5.0

```
prop.table(table(pre$correct))
```

```
##
##     False      True
## 0.2294372 0.7705628
```

Visualizeation(Probabilty Density Function Graph)

** #####From the patient’s point of view, I visualized the diabetic results in probability #####Density graph with patients diagnosis with diabetes showstrong line included, so #####that they can check their status at once.

Patient with Diabetes can be view by red line.

```
library(ggplot2)
diabetes_summary <- function(new,data) {



## [a] Reshape the new dataset for ggplot

library(reshape2)

m_train <- melt(data, id="diabetic")

m_new <- melt(new)




## [b] Save mean of Malignant value

library(dplyr)

mal_mean <- subset(data, diabetic=="Yes", select=-9)

mal_mean <- apply(mal_mean,2,mean)



## [c] highlight with red colors line

library(stringr)

mal_col <- ifelse((round(m_new$value,3) > mal_mean), "red", "black")







## [d] Save titles : Main title, Patient Diagnosis



title <- paste("Diabetic Diagnosis Plot")




## ???[f] View plots highlighting values above average of malignant patient
```

```
res_mean <- ggplot(m_train, aes(x=value,color=diabetic, fill=diabetic))+
    geom_histogram(aes(y=..density..), alpha=0.5, position="identity", bins=50)+
    geom_density(alpha=.2)+
    scale_color_manual(values=c("#15c3c9","#f87b72"))+
    scale_fill_manual(values=c("#61d4d6","#f5a7a1"))+
    geom_vline(data=m_new, aes(xintercept=value),
               color=mal_col, size=1.5)+
    geom_label(data=m_new, aes(x=Inf, y=Inf, label=round(value,3)), nudge_y=2,
               vjust = "top", hjust = "right", fill="white", color="black")+
    labs(title=title)+
    theme(plot.title = element_text(face='bold', colour='black', hjust=0.5, size=15))+
    theme(plot.subtitle=element_text(lineheight=0.8, hjust=0.5, size=12))+
    labs(caption="[Training 537 Pima Indians Diabetes Data]")+
    facet_wrap(~variable, scales="free", ncol=4)




## [g] output graph
res_mean


}
```

Testing Function

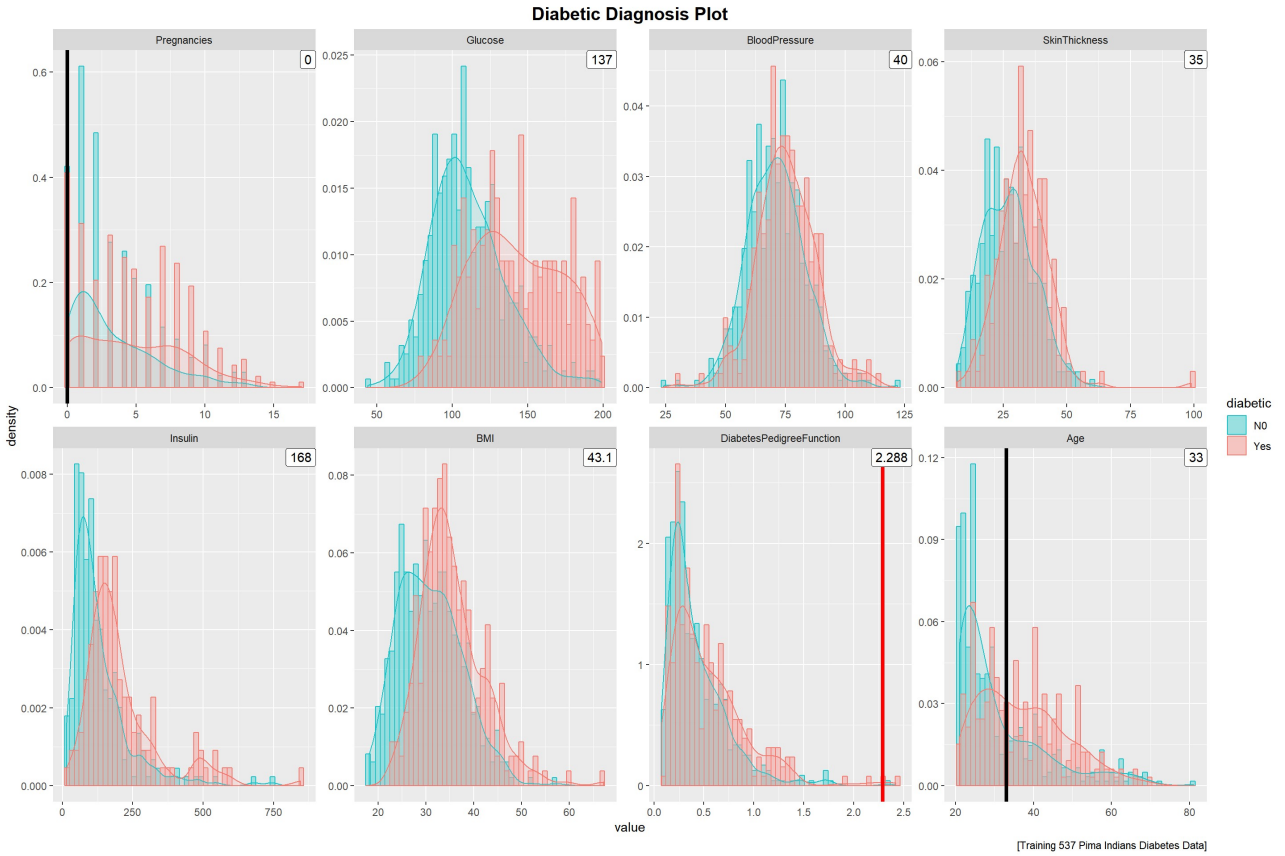
```
diabetes_summary(Y, mydata)
```

```
## Warning: package 'reshape2' was built under R version 3.4.4
```

```
## No id variables; using all as measure variables
```

```
## Warning: Removed 652 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 652 rows containing non-finite values (stat_density).
```




```
diabetes_summary(N, mydata)
```

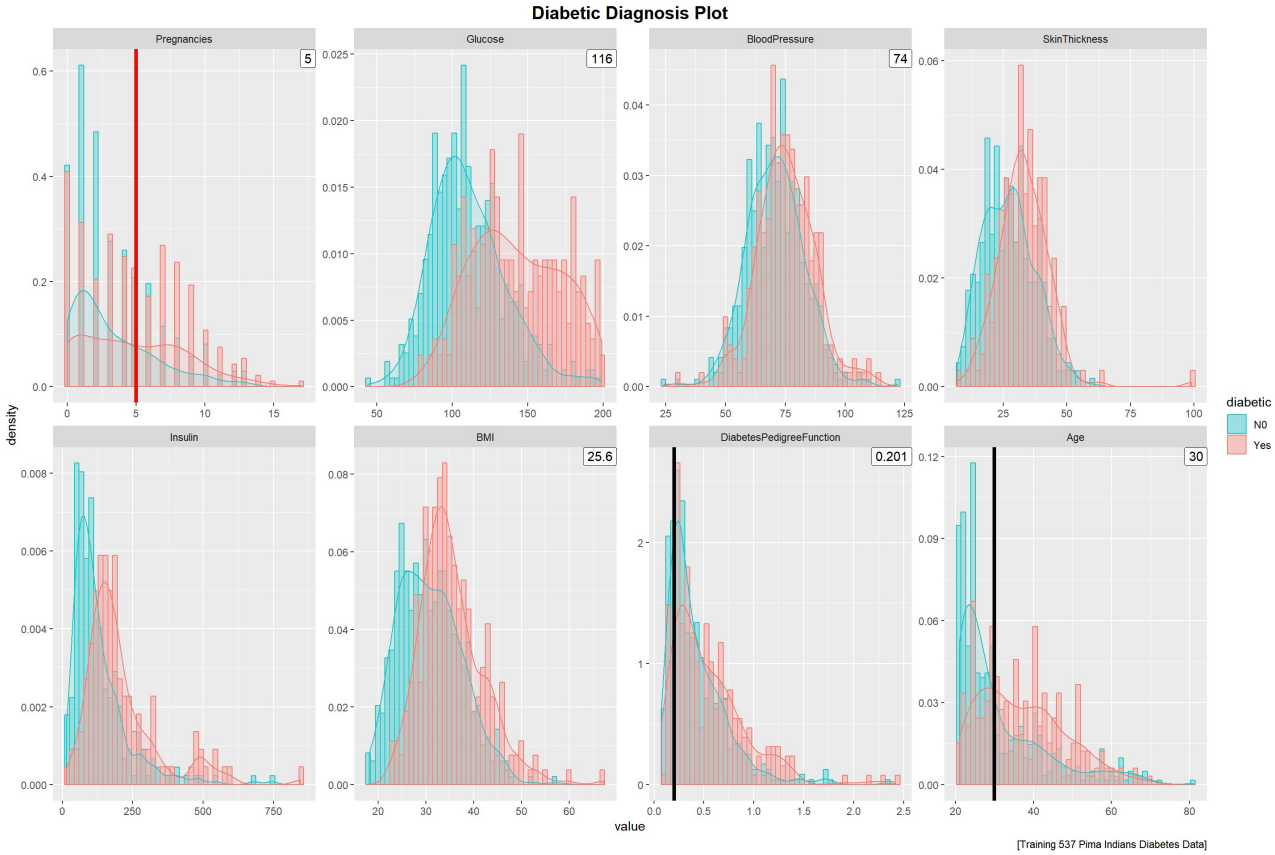
```
## No id variables; using all as measure variables
```

```
## Warning: Removed 652 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 652 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 2 rows containing missing values (geom_vline).
```

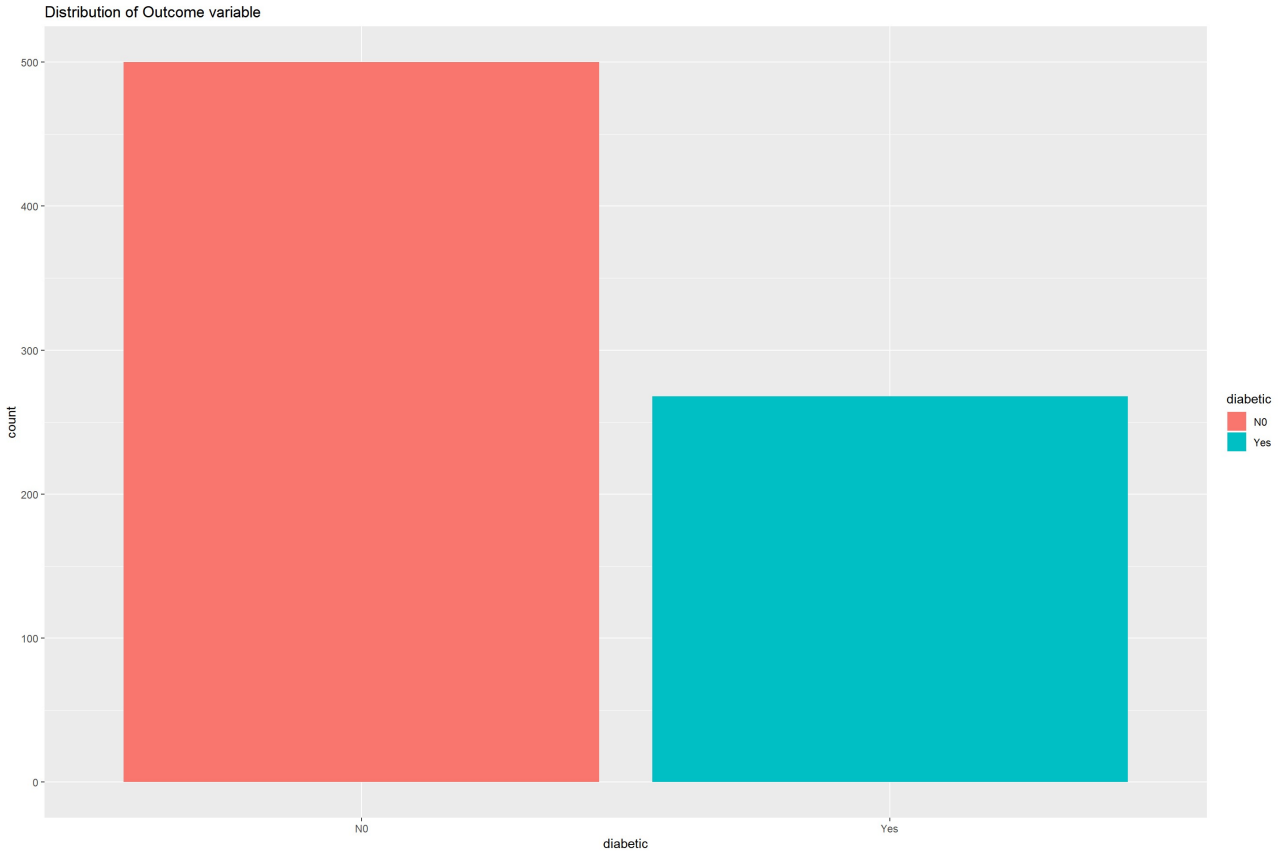
```
## Warning: Removed 2 rows containing missing values (geom_label).
```



Finding relation between BMI and Diabetic.

```
library(ggplot2)
mydata$diabetic<- factor(mydata$diabetic)

# 1. Outcome
ggplot(mydata,aes(diabetic,fill = diabetic)) +
  geom_bar() +
  ggtitle("Distribution of Outcome variable")
```




```
picture2 <- ggplot(mydata, aes(BMI, fill = diabetic)) +
  geom_histogram() +
  theme(legend.position = "bottom") +
  ggtitle("Variation of BMI of women Vs Diabetic")

picture1 <- ggplot(mydata, aes(x = diabetic, y = BMI,fill = diabetic)) +
  geom_boxplot(binwidth = 5) +
  theme(legend.position = "bottom") +
  ggtitle("Variation of BMI of women Vs Diabetic")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
install.packages('gridExtra', dependencies=TRUE, repos='http://cran.rstudio.com/')
```

```
## Installing package into 'C:/Users/ADMIN/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
```

```
## package 'gridExtra' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\ADMIN\AppData\Local\Temp\RtmpQ1NeAp\downloaded_packages
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.4
```

```
##
## Attaching package: 'gridExtra'
```

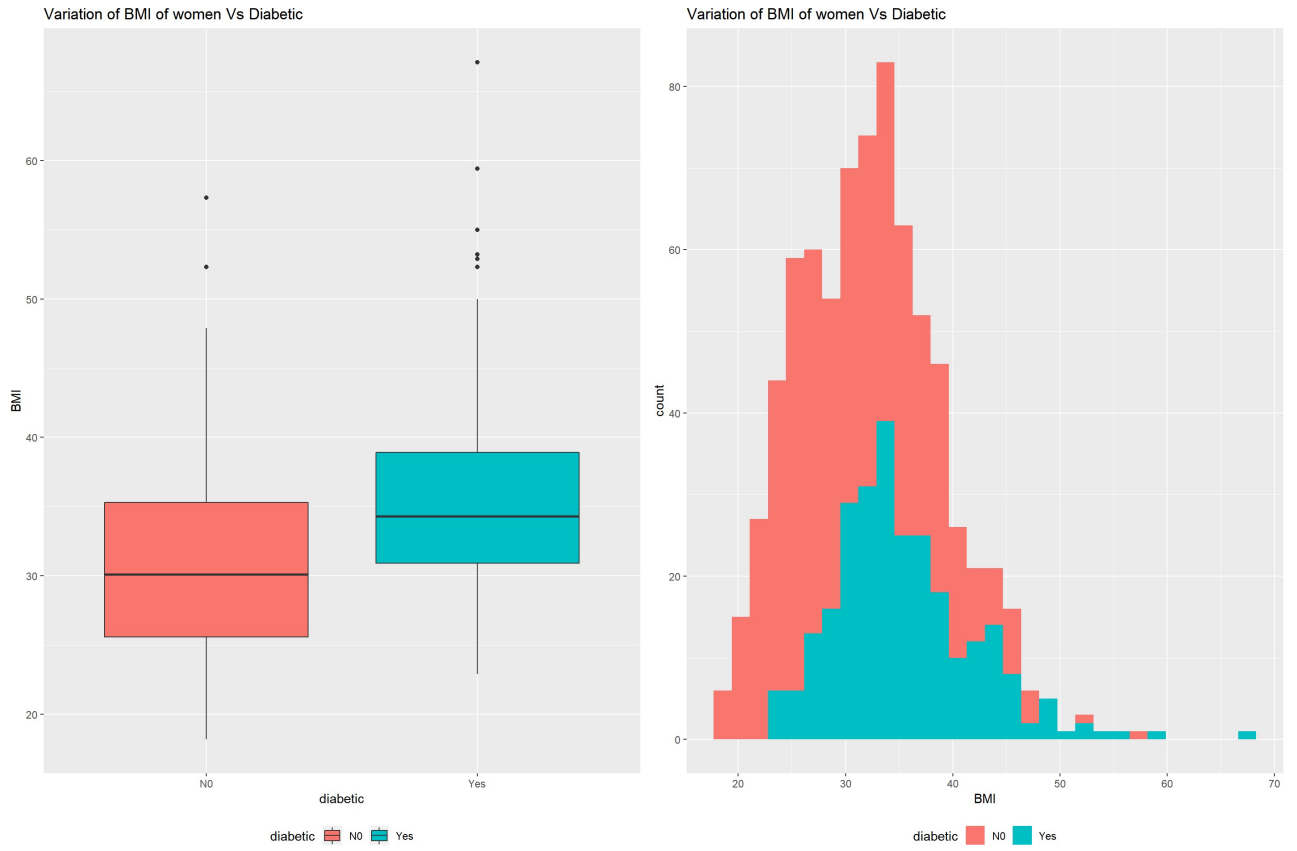
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
gridExtra::grid.arrange(picture1, picture2, ncol = 2)
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 11 rows containing non-finite values (stat_bin).
```



We will compare women with diabetes with high BMI and BP Values.

Patients with low values of BMI and Skin Thickness did not have Diabetese.

```
library(ggplot2)
picture1 <- ggplot(mydata,aes(x=BMI,y=BloodPressure))+
  geom_point(aes(color=diabetic))+
  theme(legend.position = "bottom") +
  ggtitle("Relationship of BMI with BP Vs Diabetic")

picture2 <- ggplot(mydata,aes(x=BMI,y=SkinThickness))+
  geom_point(aes(color=diabetic))+
  theme(legend.position = "bottom") +
  ggtitle("Relationship of BMI with Skin Thickness Vs Diabetic")
install.packages('gridExtra', dependencies=TRUE, repos='http://cran.rstudio.com/')
```

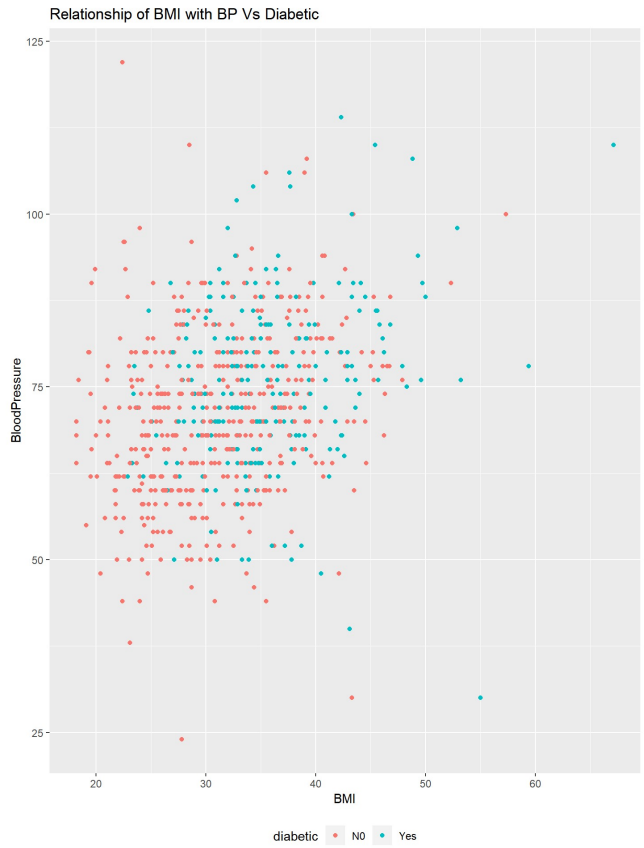
```
## Installing package into 'C:/Users/ADMIN/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
```

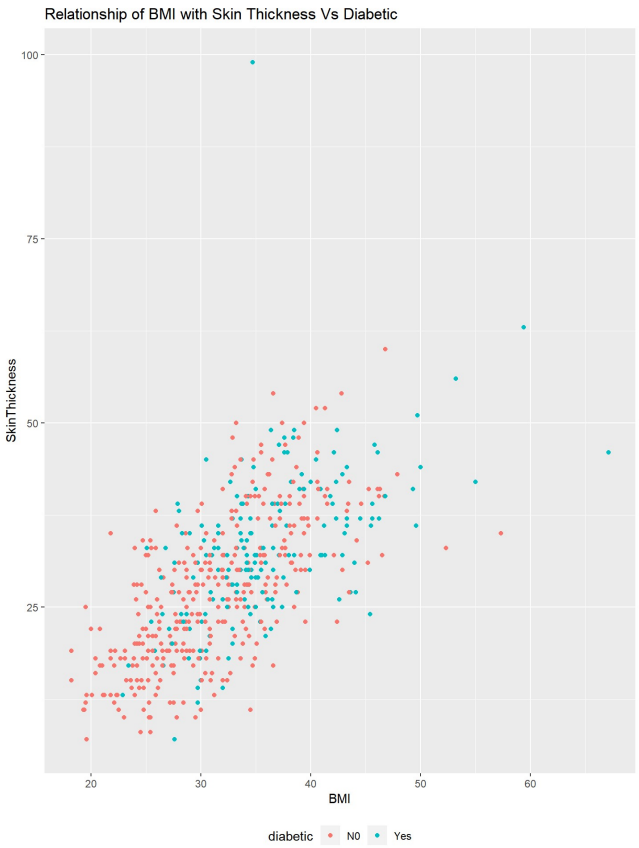
```
## Warning: package 'gridExtra' is in use and will not be installed
```

```
library(gridExtra)
gridExtra::grid.arrange(picture1, picture2, ncol = 2)
```

```
## Warning: Removed 39 rows containing missing values (geom_point).
```

```
## Warning: Removed 229 rows containing missing values (geom_point).
```





Conclusion

After applying many algorithms and feature manipulation and We got the best accuracy of Tune C5.0 is 77% accuracy and we done lot of exploratory data analysis to come with conclusion that their are few patients diabetic and few are not diabetic based on information in the dataset.

In real life not alwayas high acuuracy model is perfect so we can also use prune in which show 70 percentage accuracy.BMI , skinthickness and Insulin is 40% correlated.

And age and pregnancies are 50% correlated

As well as as more the weight / BMI of the patients more chances of diabetes .