بررسی مجموعه داده های سرطان سینه ویسکانسین ـ پروژه درس داده کاوی گروه ریاضی و آمار دانشگاه اراک

نویسندگان:آتوسا رستمی و محراب عتیقی

زمستان۱۴۰۰ ـ نیمسال۰۰۱

فهرست مطالب

۱ مقدمه

سرطان سینه یکی از شایع ترین ها در کنارسرطان ریه ونایژه،سرطان پروستات ،سرطان روده بزرگ و سرطان پانکراس است که ۱۵درصد از کل موارد جدید سرطان را تنها در ایالت متحده تشکیل می دهد و این یک موضوع تحقیقاتی بسیار عالی است. استفاده از علم داده و رویکردهای یادگیری ماشین در زمینه های پزشکی بسیار پر بار است و کمک بزرگی در فرآیندتصمیم گیری و تشخیص به شمار میرود. پزشکان با مشاهده روند افزایشی تاسف بار سرطان سینه به داده های زیادی دست می یابند که در پیشبرد تحقیقات بالینی و پزشکی کاربرد قابل توجهی داردکه در اینجاکاربرد علم داده ویادگیری ماشین در دامنه فوق الذکر بیشتر مورد توجه قرار میگیرد. در این گزارش ما قصد داریم با استفاده از الگوریتم های بردارهای ماشین پشتیبان ،درخت های رگرسیونی،knn، به طبقه بندی سرطان سینه بپردازیم و دقت مدل های برآورد شده را گزارش دهیم.

۲ معرفی مجموعه داده ها

مجموعه داده مورد بررسی در این گزارش مجموعه داده های سرطان سینه ویسکانسین میباشد.

(WDBC)=Wisconsin Breast Cancer Dataset

این مجموعه داده از تصویر برداری های دیجیتالی که توسط یک سوزن ظریف که مایع درون سلولی، سلول های سرطان سینه رابیرون میکشد ثبت شده است. به این روش در علم پزشکی به اختصار (FNA) گفته میشود. این داده ها ویژگی های هسته های سلولی موجود در تصاویر را توصیف میکنند.

۱.۲ تست FNA چیست؟

نمونه گیری سوزنی یا همان FNA یک نوع عمل بیوپسی یا همان نمونه برداری از بافت است. در نمونه گیری سوزنی یک سوزن نازک در ناحیه ای از بافت ناهنجار ظاهر شده برای نمونه گیری وارد میشود. همانند سایر بیوپسی ها،نمونه جمع آوری شده در سوزن مرغوب می تواند به تشخیص کمک کند یا شرایط مانند سرطان را پیشبینی کند این روش نمونه گیری معمولا ایمن محسوب میشود و عوارض جانبی آن نادر است.

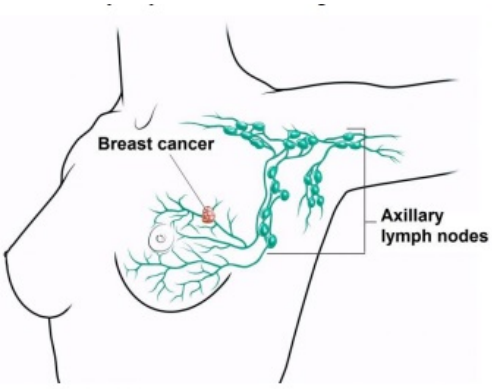
۲.۲ چه زمانی ازFNA استفاده میشود؟

این مدل از نمونه گیری اغلب در نواحی متورم یا توده هایی که دقیقا زیر پوست قرار دارند انجام میشود.این توده ها می توانند در زمان معاینه پزشک یا در زمان انجام یکی از خدمات تصویر برداری مشاهده شوند.این آزمایشات عبارتند از:

۱. تصویر برداری سیتی اسکن
۲. ماموگرافی
۳. سونوگرافی

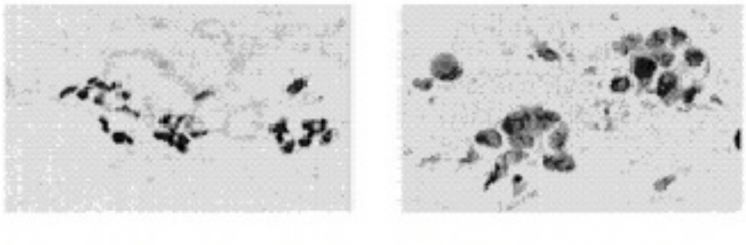
که ممکن است نقاط ناهنجار در اعماق بدن را نیز کشف کنند. به همین دلیل پزشکان این نمونه برداری را از مناطق کیست ها(توده های پر از مایعات) ، برآمدگی ها یا کپه ها(توده های جامد)، غدد لنفاوی بزرگ شده را تجویز می کنند.

در این مجموعه داده نمونه برداری از کیست ها مشکوک نزدیک به غدد لنفاوی زیر بغل که اولین مکانی است که احتمال گسترش سرطان سینه وجود دارد انجام شده واطلاعات هسته های سلولی موجود در تصاویر را ثبت نموده اندکه در تصویر زیر محل دقیق وقوع سرطان و نمونه گیری مشخص شده است.



شکل ۱: Axillary lymph nodes near a breast with cancer

حال با استفاده از اطلاعات ثبت شده برای تشخیص غدد سرطانی بدخیم و نمونه های خوش خیم غیر سرطانی استفاده خواهیم نمود. در شمل زیر نیز نوع نمونه گیری و تصویر برداری دیجیتالی نشان داده شده است:



شکل ۲: Digitized image of FNA

یک گام مهم در مدل تشخیص سرطان سینه استخراج ویژگی است. متغیر های بهینه باید ویژگی های موثر ومتمایز زیادی داشته باشند در حالی که ویژگی های فضای اطراف سلولی را باید بیشتر کاهش دهند با مشقت بعد چندی نیز نشان داده میشود که تراکم نمونه گیری از دادهای آموزشی نیز آنقدر پایین است که نمی توان تخمین معنا داری از ابعاد بالا را وعده داد و تابع طبقه بندی با تعداد محدودی از داده های آموزشی موجود است.

۳.۲ جزئیات ویژگی های موجود در این مجموعه داده

جدول ۱: Table-1

...	ویژگی	حوزه
۱	کد نمونه اخذ شده	id
۲	ضخامت توده	۱–۱۰
۳	یکنواختی اندازه سلول	۱–۱۰
۴	یک نواختی شکل سلول	۱–۱۰
۵	چسبندگی حاشیه ای	۱–۱۰
۶	اندازه تک سلول پوششی	۱–۱۰

...	ویژگی	حوزه
۷	هسته بدون پوشش سلول	۱۰–۱
۸	کروماتین ملایم	۱۰–۱
۹	هسته طبیعی	۱۰–۱
۱۰	میتوز ها	۱۰–۱
۱۱	رده ها	۰خوش خیم ۱ بدخیم

۱.۳.۲ مشاهدات موارد بدخیم

درقطر مسدود شده، سلول های نرمال تمایل دارند در دسته های تک لایه ای گروه بندی شوند در حالی که اغلب سلول های سرطانی در دسته های چند لایه ای گروه بندی میشوند. سلول های نرمال در شکل و اندازه هم یکنواختی دارند در حالی که سلول های سرطانی تمایل دارند اندازه و شکل متفاوتی داشته باشند به همین دلیل این پارامتر ها در تعیین وتشخیص سرطانی بودن یانبودن یک سلول بسیار ارزشمند هستند. سلول ها در حالت عادی تمایل دارند بهم بچسبند در حالی که سلول های سرطانی تمایل دارند این توانایی را از دست بدهند بنابراین از دست دادن چسبندگی نشانه بدخیمی سرطان است. همینطور سلول های اپتیال که به طور قابل توجهی بزرگ شده اند ممکن است یک سلول بدخیم باشند. هسته بدون پوشش سلولی اصطلاحی است که برای هسته هایی استفاده میشود که توسط سیتوپلاسم احاطه نشده اندو خبر از وقوع سرطان می دهند.

۲.۳.۲ مشاهدات موارد خوش خیم

کرماتین ملایم یک بافت یک نواخت از هسته را که دیده میشود، توصیف میکند در سلول های مشکوک به سرطان کروماتین درشت تر است. در سلول های طبیعی ساختار هسته معمولا بسیار کوچک است اما در سلول های سرطانی هسته برجسته تر می شود و گاهی اوقات تعداد آنها بیشتر می شود. ودر نهایت میتوز هسته ای یکی از عوامل مهمی است که پاتولوژیست ها می توانند با شمارش آن خوش خیم یا بد خیم بودن سرطان را تعیین کنند که طبیعتا سلول های سرطانی خوش خیم تقسیمات سلولی کمتری دارند.

۳.۳.۲ معرفی داده های ثبت شده در مجموعه داده (WDBC)

۱. ID= شماره شناسایی نمونه
۲. diagnosis = B , M = تشخیص، خوش خیم و بد خیم(متغیر پاسخ)

متغیر های پیشگو ۱۰ ویژگی با ارزش های واقعی برای هر هسته سلول محاسبه شده به صورت زیر هستند:

۳. شعاع
۴. محیط بافت
۵. مساحت
۶. صافی
۷. فشردگی
۸. تقعر
۹. نقاط مقعر و تقارن
۱۰. بعد فراکتال (فراکتال ها اشکال هندسی و چند جزئی هستند که اگر آن ها را به چند قسمت تقسیم کنیم، هر قسمت کوچک شده، کپی و برابر کل شکل است.)

این ۱۰ مشاهده به صورت میانگین و خطای استاندارد و بدترین و بزرگ ترین میانگین ذخیره شدخ اند.

(mean = پسوند میانگین se = پسوند خطای استاندارد worst = پسوند بدترین و بزرگ ترین میانگین)

۳ مصور سازی

به عنوان مرحله اول داده کاوی باید مصور سازی را اجرا کنیم به این دلایل:

۱) پاکسازی داده ها و پیدا کردن مقادیر نادرست
۲) یافتن داده های گمشده
۳) یافتن سطر های تکراری یا ستون های یکسان
۴) انتخاب متغیر یا مشتقات متغیر ها
۵) تعیین اندازه مناسب گروه ها

۶) کاهش بعد و تلفیق رسته ها
۷) تعیین متغیر های مورد جمع آوری

۱.۳ مصور سازی داده های (WDBC)

۱.۱.۳ بازرسی بصری مجموعه داده

```
#############libraries###############
library("ggplot2")
library("caTools")
```

```
## Warning: package 'caTools' was built under R version 4.1.2
```

```
library("corrplot")
```

```
## Warning: package 'corrplot' was built under R version 4.1.2
```

```
## corrplot 0.92 loaded
```

```
library("dplyr")
```

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

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

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

```
#Load dataset and DATA EXPLORATION
wdbc=read.csv("C:/Users/atusa/Downloads/data.csv",header=TRUE)
head(wdbc,3)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38          122.8      1001
## 2   842517         M       20.57        17.77          132.9      1326
## 3 84300903         M       19.69        21.25          130.0      1203
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
```

```
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
##   symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1     0.03003             0.006193        25.38         17.33           184.6
## 2     0.01389             0.003532        24.99         23.41           158.8
## 3     0.02250             0.004571        23.57         25.53           152.5
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1       2019           0.1622            0.6656          0.7119
## 2       1956           0.1238            0.1866          0.2416
## 3       1709           0.1444            0.4245          0.4504
##   concave.points_worst symmetry_worst fractal_dimension_worst  X
## 1               0.2654         0.4601                 0.11890 NA
## 2               0.1860         0.2750                 0.08902 NA
## 3               0.2430         0.3613                 0.08758 NA
```

```
glimpse(wdbc)
```

```
## Rows: 569
## Columns: 33
## $ id                      <int> 842302, 842517, 84300903, 84348301, 84358402, ~
## $ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
## $ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
## $ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
## $ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
## $ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
## $ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
## $ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
## $ concave.points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
## $ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
## $ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
## $ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
## $ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
## $ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
## $ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
## $ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
## $ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
## $ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
## $ concave.points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
## $ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
## $ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
## $ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
## $ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 27.6~
## $ perimeter_worst         <dbl> 184.60, 158.80, 152.50, 98.87, 152.20, 103.40,~
## $ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
## $ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
## $ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
```

```
## $ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
## $ concave.points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
## $ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
## $ X                       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```
#structure of the dataset
str(wdbc)
```

```
## 'data.frame':    569 obs. of  33 variables:
##  $ id                      : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844
##  $ diagnosis               : chr  "M" "M" "M" "M" ...
##  $ radius_mean             : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean            : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean          : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean               : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean         : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean        : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean          : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean     : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean           : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean  : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se               : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se              : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se            : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                 : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se           : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_se          : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se            : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se       : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se             : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se    : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ radius_worst            : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst           : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst         : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst              : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst        : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst       : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst         : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst    : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst          : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
##  $ X                       : logi  NA NA NA NA NA NA ...
```

```
#dimension of data set
dim(wdbc)
```

```
## [1] 569  33
```

```
#summary of data set
summary(wdbc)
```

```
##        id              diagnosis          radius_mean      texture_mean
##  Min.   :     8670   Length:569         Min.   : 6.981   Min.   : 9.71
##  1st Qu.:   869218   Class :character   1st Qu.:11.700   1st Qu.:16.17
##  Median :   906024   Mode  :character   Median :13.370   Median :18.84
##  Mean   : 30371831                      Mean   :14.127   Mean   :19.29
```

```
##  3rd Qu.:  8813129                     3rd Qu.:15.780   3rd Qu.:21.80
##  Max.   :911320502                     Max.   :28.110   Max.   :39.28
##  perimeter_mean     area_mean      smoothness_mean   compactness_mean
##  Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938
##  1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492
##  Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263
##  Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434
##  3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040
##  Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540
##  concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154
##  Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
##    radius_se        texture_se      perimeter_se       area_se
##  Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
##  1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
##  Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
##  Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
##  3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
##  Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
##  smoothness_se      compactness_se      concavity_se     concave.points_se
##  Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638
##  Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930
##  Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796
##  3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710
##  Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790
##   symmetry_se       fractal_dimension_se  radius_worst   texture_worst
##  Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
##  1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
##  Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
##  Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
##  3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
##  Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
##  perimeter_worst    area_worst     smoothness_worst  compactness_worst
##  Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729
##  1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720
##  Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190
##  Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427
##  3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910
##  Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800
##  concavity_worst  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.2267   Median :0.09993      Median :0.2822   Median :0.08004
##  Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
##     X
##  Mode:logical
##  NA's:569
```

```
##
##
##
##
```

```
##remove na's
wdbc=wdbc[-33]
summary(wdbc)
```

```
##        id             diagnosis          radius_mean      texture_mean
##  Min.   :     8670   Length:569         Min.   : 6.981   Min.   : 9.71
##  1st Qu.:   869218   Class :character   1st Qu.:11.700   1st Qu.:16.17
##  Median :   906024   Mode  :character   Median :13.370   Median :18.84
##  Mean   : 30371831                      Mean   :14.127   Mean   :19.29
##  3rd Qu.:  8813129                      3rd Qu.:15.780   3rd Qu.:21.80
##  Max.   :911320502                      Max.   :28.110   Max.   :39.28
##  perimeter_mean     area_mean      smoothness_mean   compactness_mean
##  Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938
##  1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492
##  Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263
##  Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434
##  3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040
##  Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540
##  concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154
##  Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
##    radius_se        texture_se      perimeter_se       area_se
##  Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
##  1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
##  Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
##  Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
##  3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
##  Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
##  smoothness_se      compactness_se      concavity_se     concave.points_se
##  Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638
##  Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930
##  Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796
##  3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710
##  Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790
##   symmetry_se       fractal_dimension_se  radius_worst   texture_worst
##  Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
##  1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
##  Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
##  Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
##  3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
##  Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
##  perimeter_worst    area_worst     smoothness_worst  compactness_worst
##  Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729
##  1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720
##  Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190
```

```
##  Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427
##  3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910
##  Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800
##  concavity_worst  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   Min.   :0.05504
##  1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1st Qu.:0.07146
##  Median :0.2267   Median :0.09993      Median :0.2822   Median :0.08004
##  Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   Mean   :0.08395
##  3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3rd Qu.:0.09208
##  Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   Max.   :0.20750
```

۲.۱.۳ تحلیل داده ها

تعداد زنان مبتلا در مرحله خوش خیم و بدخیم:

```
wdbc %>% count(diagnosis)
```

```
##   diagnosis   n
## 1         B 357
## 2         M 212
```

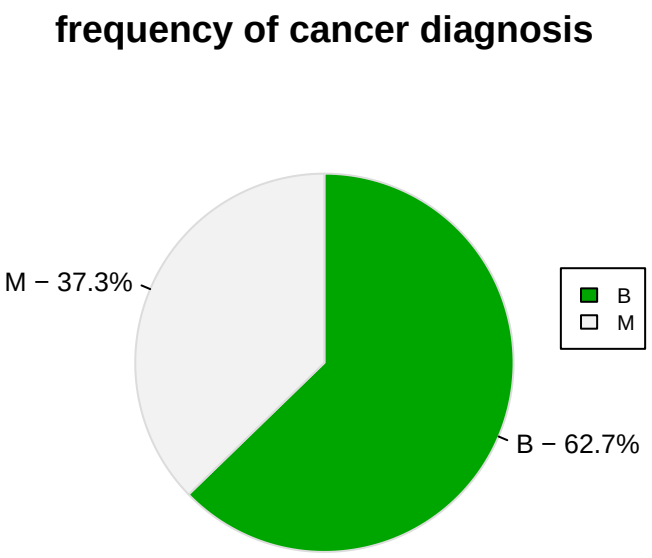
درصد زنان مبتلا در مرحله خوش خیم و بدخیم:

```
wdbc %>% count(diagnosis)%>%group_by(diagnosis) %>%
  summarize(perc_dx = round((n / 569)* 100, 2))
```

```
## # A tibble: 2 x 2
##   diagnosis perc_dx
##   <chr>       <dbl>
## 1 B            62.7
## 2 M            37.3
```

۳.۱.۳ فراوانی تشخیص سرطان با نمودار دایره ای

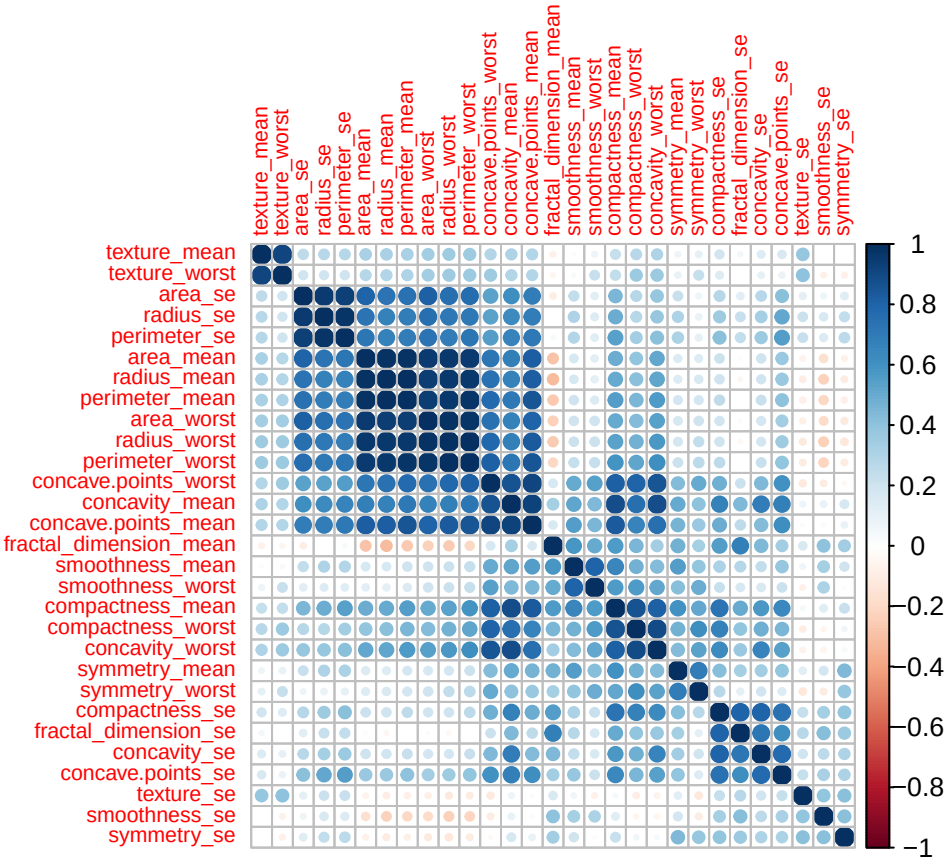
```
diagnosis.table -< table(wdbc$diagnosis)
colors -< terrain.colors(2)
 chart pie a Create #
diagnosis.prop.table -< prop.table(diagnosis.table)*100
diagnosis.prop.df -< as.data.frame(diagnosis.prop.table)
pielabels -< sprintf("1f%s.%3 - "%s, diagnosis.prop.df[,1], diagnosis.prop.table, "%")
pie(diagnosis.prop.table,
  labels=pielabels,
  clockwise=TRUE,
  col=colors,
  border="gainsboro",
  radius=8.0,
  cex=8.0,
  main=diagnosis" cancer of "frequency)
legend(1, .4, legend=diagnosis.prop.df[,1], = cex 7.0, = fill colors)
```



۴.۱.۳ نمودار همبستگی (نمودار حرارتی)

برای پی بردن به ارتباط بین ۳۰ متغیر پیشگو و یافتن روابط دو متغیره بین این ۳۰ پیشگو از نمودار حرارتی که میزان همبستگی را نمایش میدهد کمک میگیریم.

```
collinearity calculate #
c -< cor(wdbc[,3:31])
corrplot(c, = order "hclust", = tl.cex 7.0)
```



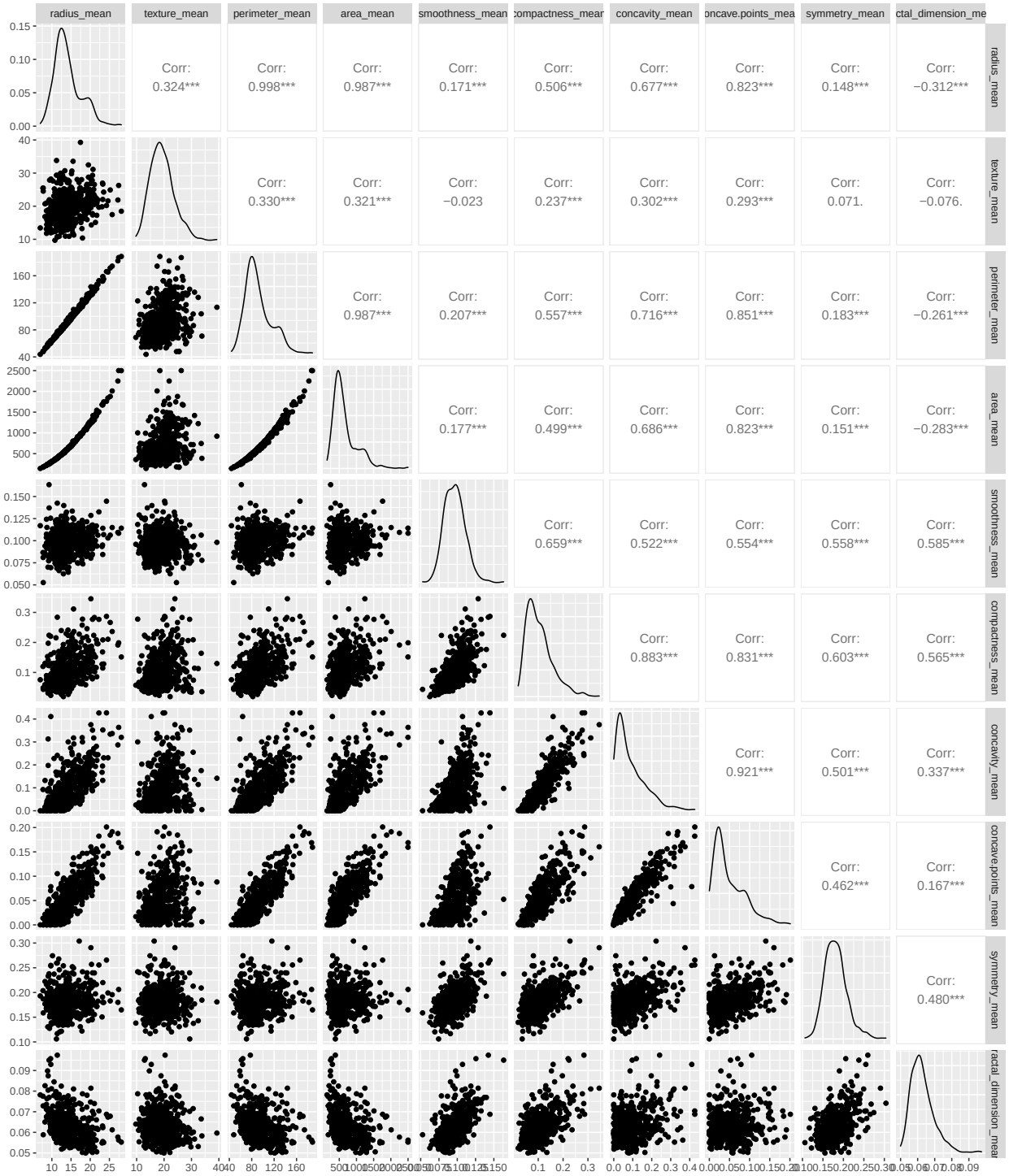
هر چقدر به سمت رنگ آبی تیره میرویم همبستگی بین متغیر ها افزایش می یابد.

۵.۱.۳ نمودار بررسی مشاهدات برای میانگین های ثبت شده متغیر ها

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

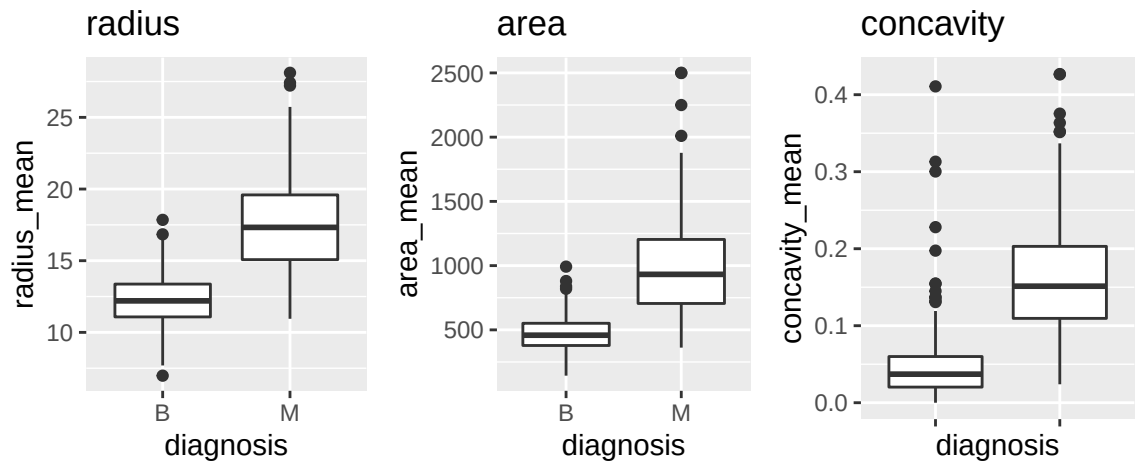
```
ggpairs(wdbc[,c(3:12)])
```



۶.۱.۳ مقایسه شعاع،مساحت،تقعر سلول های سرطانی برای تشخیص خوش خیم یا بد خیم بودن سرطان

```
library(ggpubr)
#radius
A=ggplot(data=wdbc,aes(x=diagnosis,y=radius_mean))+geom_boxplot()+ggtitle("radius")
```

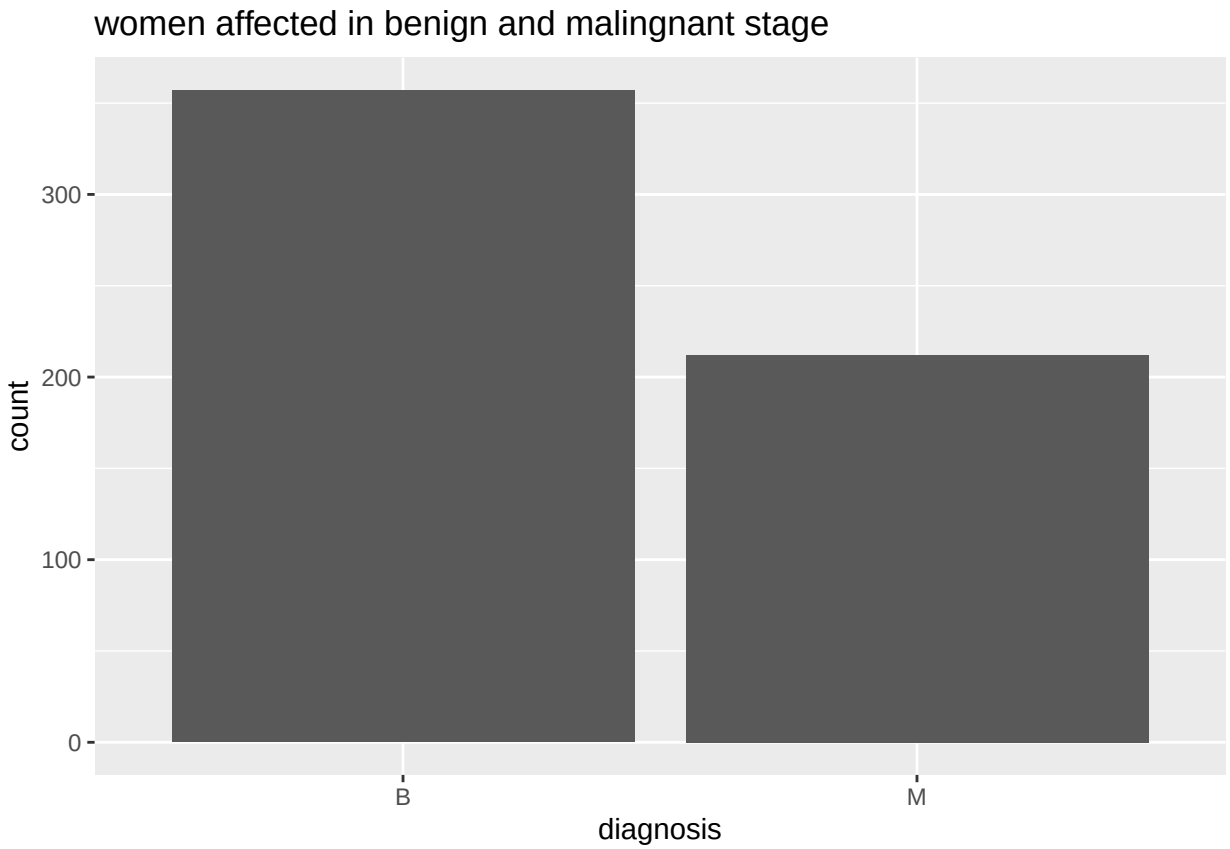
```
#area
B=ggplot(data=wdbc,aes(x=diagnosis,y=area_mean))+geom_boxplot()+ggtitle("area")

#concavity
C=ggplot(data=wdbc,aes(x=diagnosis,y=concavity_mean))+geom_boxplot()+ggtitle("concavity")

ggarrange(A, B, C + rremove("x.text"),
          ncol = 3, nrow = 1)
```



با توجه به نمودار های جعبه ای رسم شده به این موضوع پی میبریم که سلول های سرطانی بدخیم شعاع،مساحت و تقعر بیشتری نسبت به سلول های خوش خیم دارند.

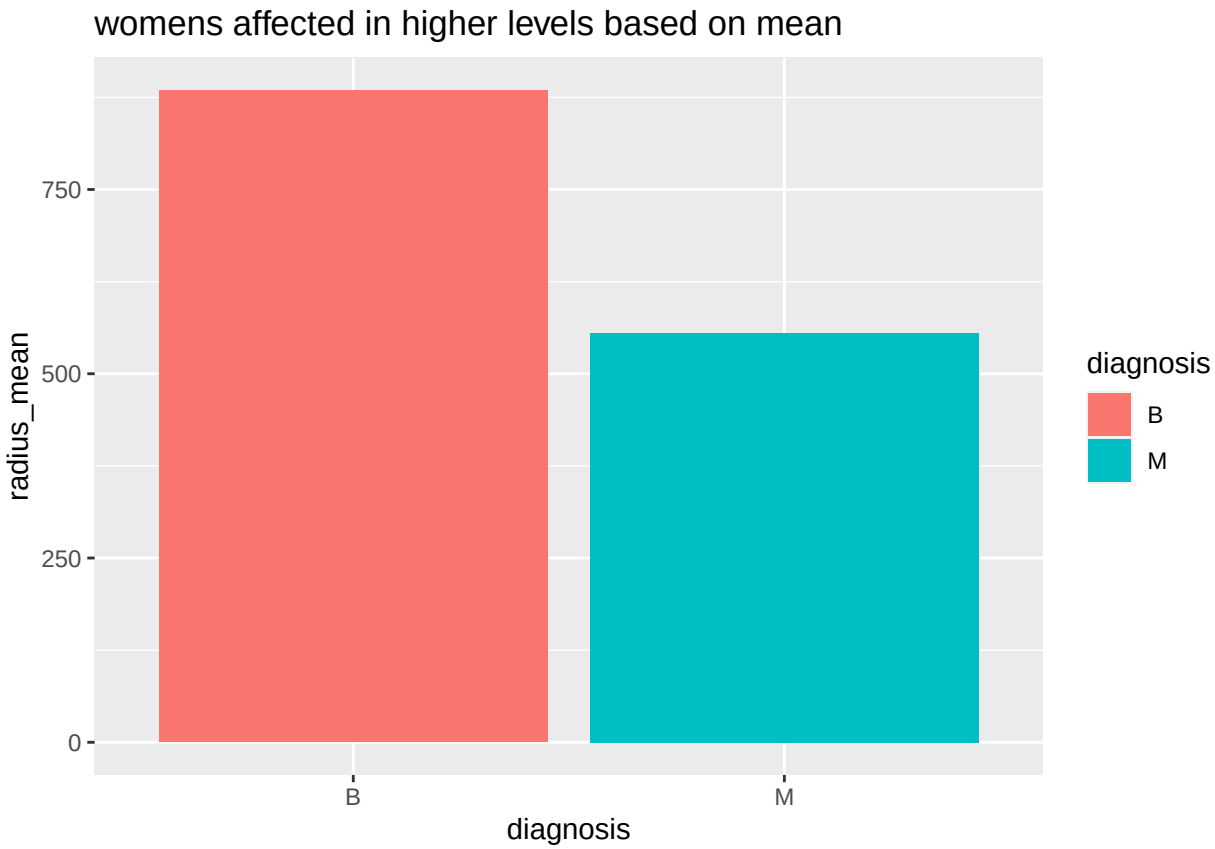
۷.۱.۳ نمودار میله ای بررسی و تحلیل مراحل برای زنان مبتلا به سرطان

```
ggplot(wdbc,aes(x=diagnosis,fill=texture_mean))+
geom_bar()+ggtitle(stage" malingnant and benign in affected "women)
```



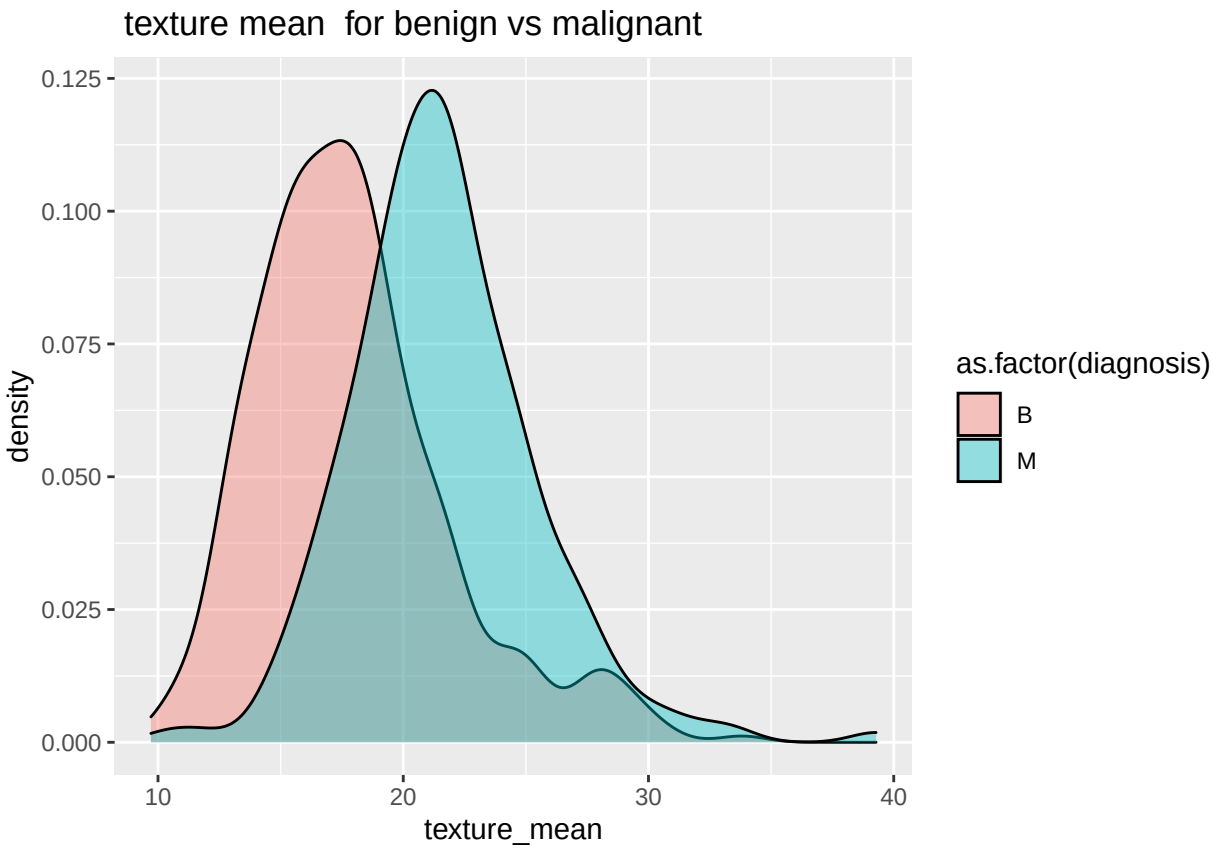
۸.۱.۳ بررسی زنان مبتلا به سرطان در سطوح بالای ابتلا بر اساس میانگین های گزارش شده در نمودار جعبه ای

```
sel_data=wdbc[wdbc$radius_mean>10&
                wdbc$radius_mean<15&
                wdbc$compactness_mean>1.0,]
ggplot(sel_data,aes(x=diagnosis,y=radius_mean,fill=diagnosis))+geom_col()+
  ggtitle(mean" on based levels higher in affected "womens)
```



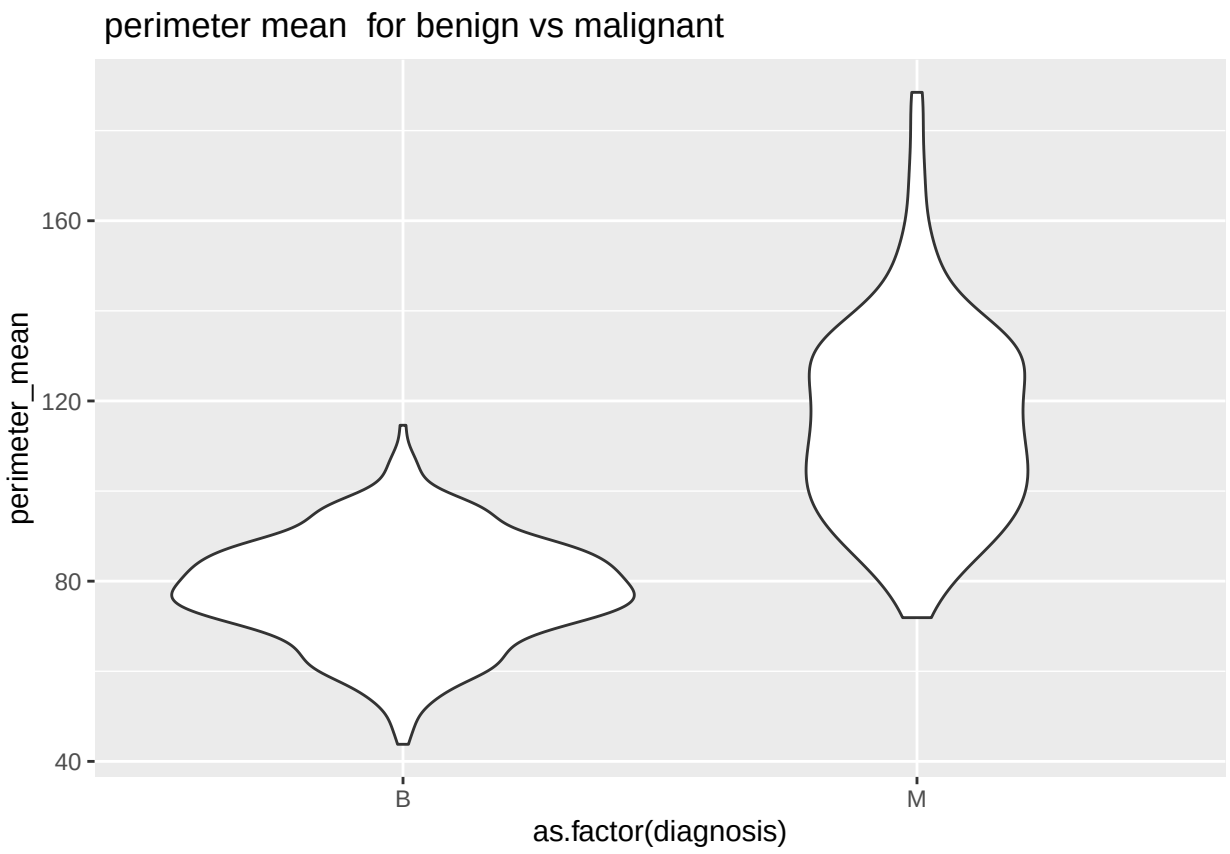
۹.۱.۳ نمودار تراکم بر اساس میانگین بافت سلولی

```
ggplot(wdbc,aes(x=texture_mean,fill=as.factor(diagnosis)))+
  geom_density(alpha=4.0)+
  ggtitle(malignant" vs benign for  mean texture ")
```



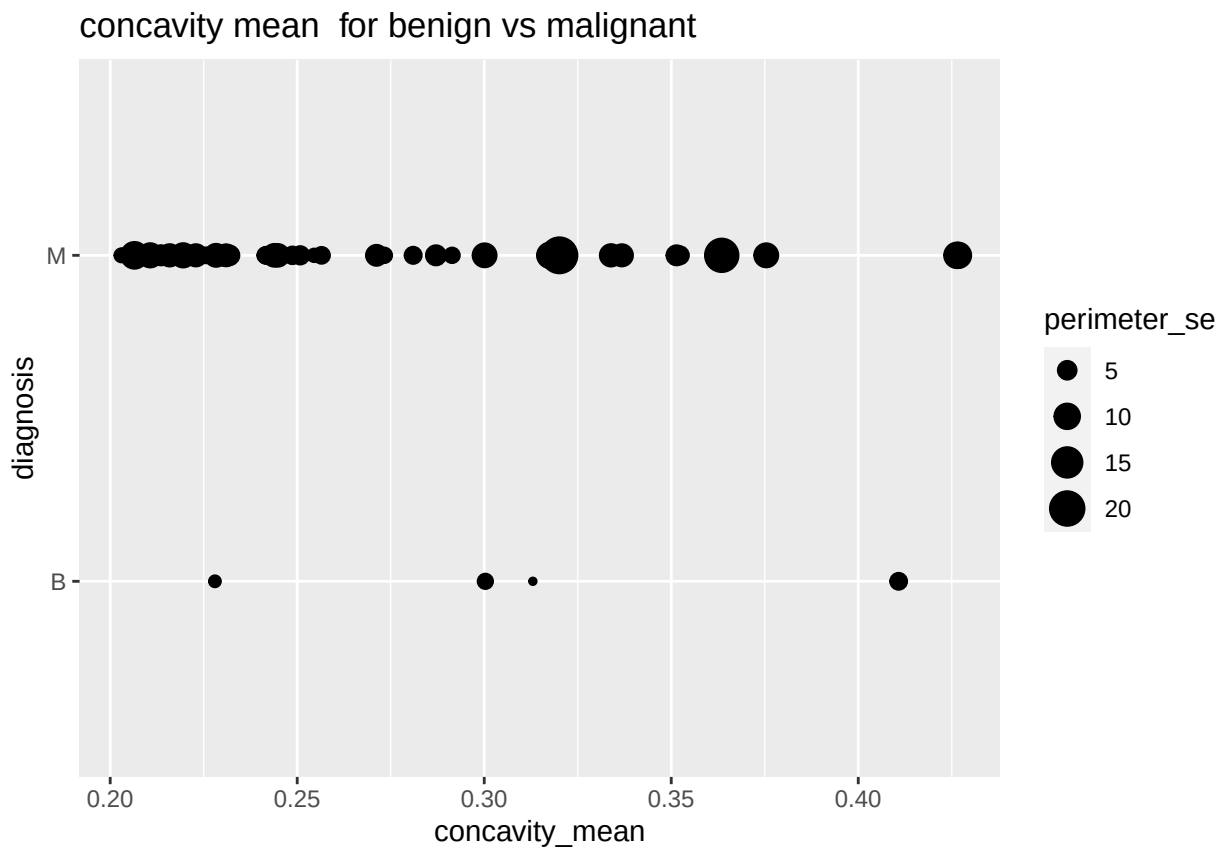
۱۰.۱.۳ تجزیه و تحلیل میانگین محیط بافت سلولی زنان مبتلا در مرحله خوش خیم و بدخیم

```
ggplot(wdbc,aes(x=as.factor(diagnosis),y=perimeter_mean))+
  geom_violin()+
  ggtitle(malignant" vs benign for  mean perimeter ")
```



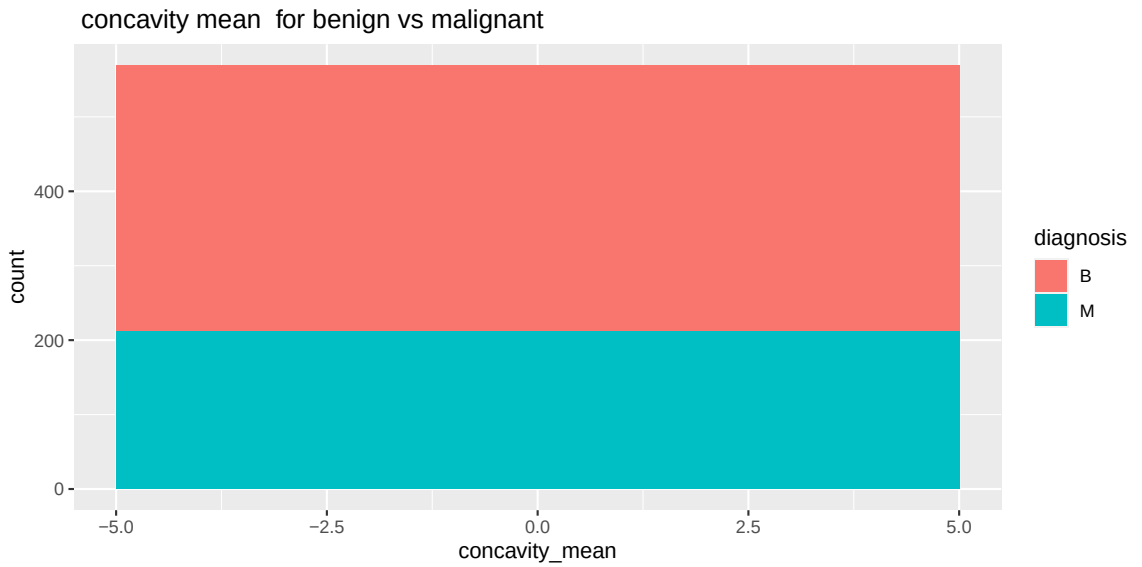
۱۱.۱.۳ تجزیه و تحلیل میانگین تقعر برای زنان مبتلا در مرحله خوش خیم و بدخیم

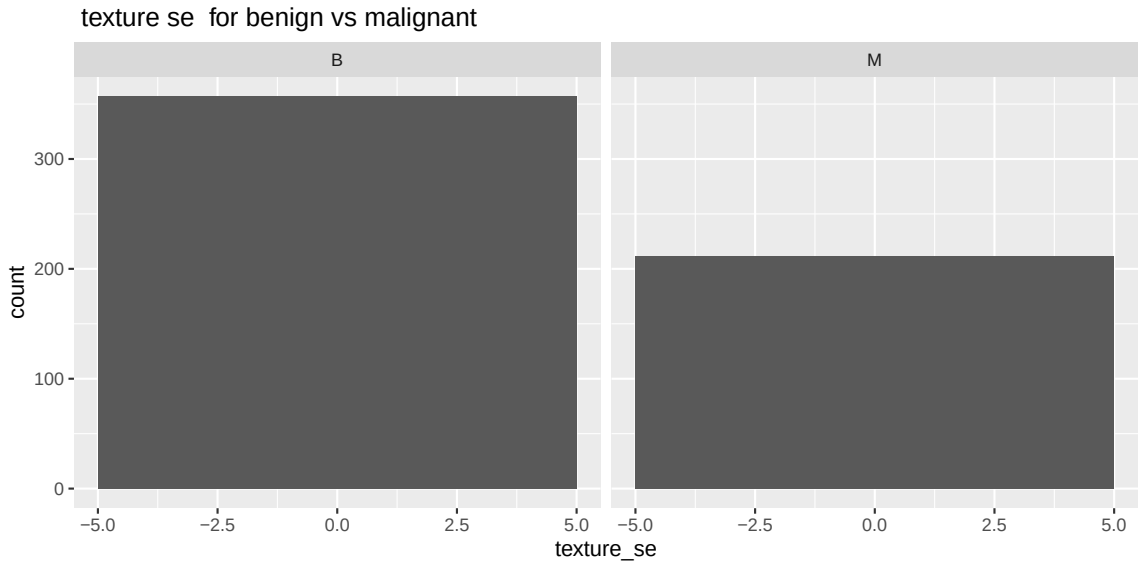
```
data1=wdbc%>%filter(concavity_mean>2.0)
ggplot(data1,aes(x=concavity_mean,y=diagnosis,size=perimeter_se))+
  geom_point()+
  ggtitle(malignant" vs benign for  mean "concavity)
```

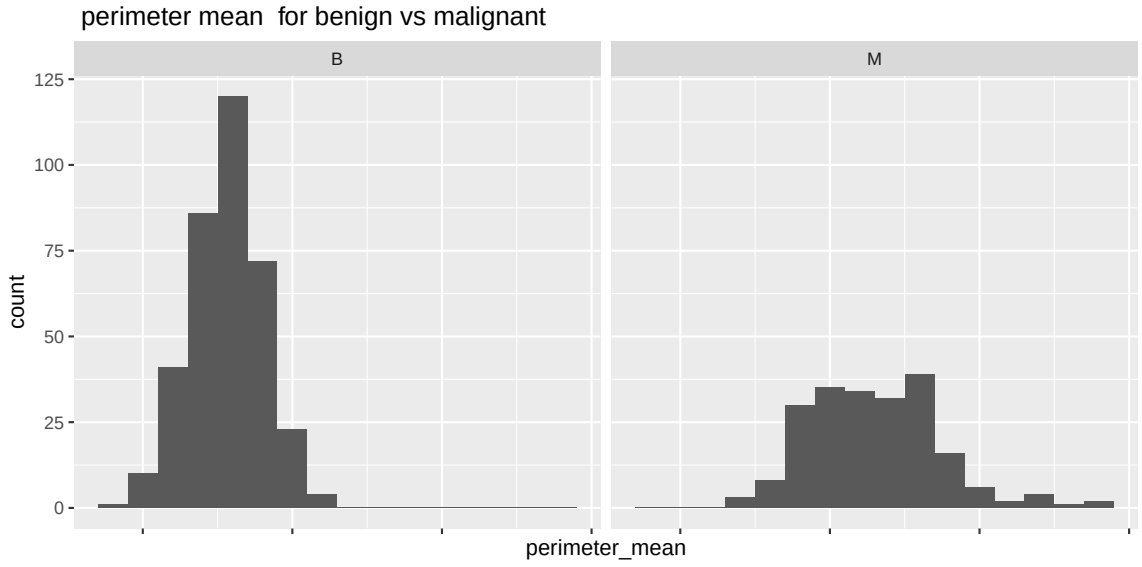


۱۲.۱.۳ تشخیص توزیع داده ها از طریق هیستوگرام

```
A=ggplot(wdbc,aes(x=concavity_mean,fill=diagnosis))+
  geom_histogram(binwidth=10)+
  ggtitle(" concavity mean  for benign vs malignant")

B=ggplot(wdbc, aes(x = texture_se)) +
  geom_histogram(binwidth=10) +
  facet_wrap(~ diagnosis)+
  ggtitle(" texture se  for benign vs malignant")

C=ggplot(wdbc, aes(x = perimeter_mean)) +
  geom_histogram(binwidth=10) +
  facet_wrap(~ diagnosis)+
  ggtitle(" perimeter mean  for benign vs malignant")

ggarrange(A, B, C + rremove("x.text"),
          ncol = 1, nrow = 3)
```


concavity mean for benign vs malignant
count
400
200
0
−5.0
−2.5
0.0
2.5
5.0
concavity_mean
diagnosis
B
M


texture se for benign vs malignant
B
M
count
300
200
100
0
−5.0
−2.5
0.0
2.5
5.0
texture_se


perimeter mean for benign vs malignant
B
M
count
125
100
75
50
25
0
perimeter_mean

باتوجه به نمودار های گزارش شده در مشاهدات بدخیم سرطان تفاوت محسوسی با مشاهدات خوش خیم دارند که مورد انتظار نیز بودند این تفاوت ها عبارتند از:

الف) داده ها دارای توزیع نامتقارن با کشیدگی مثبت و چولگی مثبت هستند به نظر می رسد توزیع داده ها نمایی باشد.

ب) میانگین بافت سلولی سرطان بدخیم بیشتر از سرطان خوش خیم است.

ج) سلول های بدخیم محیط بیشتری را اشغال می کنند.

د) سلول های سرطانی بدخیم بسیار مقعر تر از سلول های سرطانی خوش خیم هستند.

ذ) تعداد سلول های سرطانی بدخیم کمتر از سلول های سرطانی خوش خیم است.

۴ مدل بندی

۱.۴ KNN-Model

```
wdbc <- select(wdbc,-id)
dim(wdbc)
```

```
## [1] 569  31
```

```
table(wdbc$diagnosis)
```

```
##
##   B   M
## 357 212
```

```
round(prop.table(table(wdbc$diagnosis)) * 100, digits = 1)
```

```
##
##    B    M
## 62.7 37.3
```

```
#nomalized data
normalize <- function(x) {
    return ((x - min(x)) / (max(x) - min(x)))
}

new_wdbc <- as.data.frame(lapply(select(wdbc,-diagnosis), normalize))

summary(select(new_wdbc,radius_mean,smoothness_mean))
```

```
##   radius_mean     smoothness_mean
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2233   1st Qu.:0.3046
##  Median :0.3024   Median :0.3904
##  Mean   :0.3382   Mean   :0.3948
##  3rd Qu.:0.4164   3rd Qu.:0.4755
##  Max.   :1.0000   Max.   :1.0000
```

پس از پاکسازی و تبدیل داده ها، اکنون باید از داده ها برای الگوریتم های یادگیری ماشین استفاده کنیم. ما به یک مجموعه آموزشی برای ساخت مدل KNN و مجموعه تست برای بررسی دقت مدل نیاز داریم. ما از ۴۲۹ داده اول برای مجموعه داده آموزشی و ۱۴۰ داده باقیمانده برای شبیه سازی بیماران جدید استفاده خواهیم کرد.

```
wdbc_train <- new_wdbc[1:429,]
wdbc_test <- new_wdbc[430:569,]

#labels
wdbc_train_labels <- wdbc[1:429, 1]
wdbc_test_labels  <- wdbc[430:569, 1]

library(class)
```

```
## Warning: package 'class' was built under R version 4.1.2
```

```
wdbc_test_pred <- knn(train = wdbc_train, test = wdbc_test, cl = wdbc_train_labels, k = 20)
```

پس از مدلسازی داده ها در الگوریتم knn، نوبت به بررسی عملکرد مدل می رسد. ما از ماتریس درهم ریختگی برای یافتن عملکرد مدل استفاده خواهیم کرد.

```
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 4.1.2
```

```
cm =CrossTable(x = wdbc_test_labels , y = wdbc_test_pred, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  140
##
##
##                  | wdbc_test_pred
## wdbc_test_labels |         B |         M | Row Total |
## -----------------|-----------|-----------|-----------|
##                B |       104 |         1 |       105 |
##                  |     0.990 |     0.010 |     0.750 |
##                  |     0.981 |     0.029 |           |
##                  |     0.743 |     0.007 |           |
## -----------------|-----------|-----------|-----------|
##                M |         2 |        33 |        35 |
##                  |     0.057 |     0.943 |     0.250 |
##                  |     0.019 |     0.971 |           |
##                  |     0.014 |     0.236 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |       106 |        34 |       140 |
##                  |     0.757 |     0.243 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

```
cm
```

```
## $t
##    y
## x     B   M
##   B 104   1
##   M   2  33
##
## $prop.row
##    y
## x            B          M
##   B 0.99047619 0.00952381
##   M 0.05714286 0.94285714
##
## $prop.col
##    y
## x            B          M
##   B 0.98113208 0.02941176
##   M 0.01886792 0.97058824
##
## $prop.tbl
##    y
## x             B           M
##   B 0.742857143 0.007142857
##   M 0.014285714 0.235714286
```

درصد سلول ها در جدول نشان دهنده نسبت مقادیری است که در چهار دسته قرار می گیرند. در سلول بالا سمت چپ، نتایج منفی واقعی هستند. این۱۰۵ مقدار از ۱۴۰ مقدار مواردی را نشان می دهد که توده خوش خیم بود و الگوریتم kNN به درستی آن را به چنین عنوان تشخیص داد. سلول پایین سمت راست، نتایج مثبت واقعی را نشان می دهد، جایی که طبقه بندی کننده و برچسب تعیین شده بالینی موافق هستند که توده بدخیم است. در مجموع ۳۳ مورد از ۱۴۰ پیش بینی مثبت واقعی (بدخیم)بودند. مشاهدات غیر قطر اصلی نشان دهنده خطا مدل می باشد این خطا بسیار پرهزینه می باشد، زیرا ممکن است بیمار را به این باور برساند که او بدون سرطان است، در حالی که در واقع بیماری ممکن است به گسترش خود ادامه دهد که به این رویکرد بار منفی کاذب گفته میشود در مقابل این امر بار مثبت کاذب وجود دارد که بار مثبت کاذب کمتر از وضعیت منفی کاذب خطرناک است، اما می تواند بار بر مالی اضافی بر بیمار/سیستم بهداشتی و استرس اضافی بر بیمار بیافزاید.

```
(105+ 33)/ 140
```

```
## [1] 0.9857143
```

با این حال مدل دارای دقت ۹۸ درصدی است که بسیار قابل توجه است.

۲.۴ درخت تصمیم

در این مدل با minsplit=۱۱ (حداقل تعداد مشاهدات در هر تقسیم) وmaxdepth=۱۰(حداکثر عمق درخت)به بهترین نتیجه خواهیم رسید.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.5     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

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

```
library(dplyr)
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.2
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

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

```
library(corrplot)
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.1.2
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.1.2
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.2
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## randomForest 4.6-14
```

```
## 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
```

```
library(varImp)
```

```
## Warning: package 'varImp' was built under R version 4.1.2
```

```
## Loading required package: measures
```

```
## Warning: package 'measures' was built under R version 4.1.2
```

```
##
## Attaching package: 'measures'
```

```
## The following objects are masked from 'package:MLmetrics':
##
##     AUC, MAE, MAPE, MSE, RAE, RMSE, RMSLE, RRSE
```

```
## Loading required package: party
```

```
## Warning: package 'party' was built under R version 4.1.2
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'modeltools'
```

```
## The following object is masked from 'package:car':
##
##     Predict
```

```
## Loading required package: strucchange
```

```
## Warning: package 'strucchange' was built under R version 4.1.2
```

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

```
## Warning: package 'zoo' was built under R version 4.1.2
```

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

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

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 4.1.2
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.1.2
```

```
## Loaded gbm 2.1.8
```

```
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

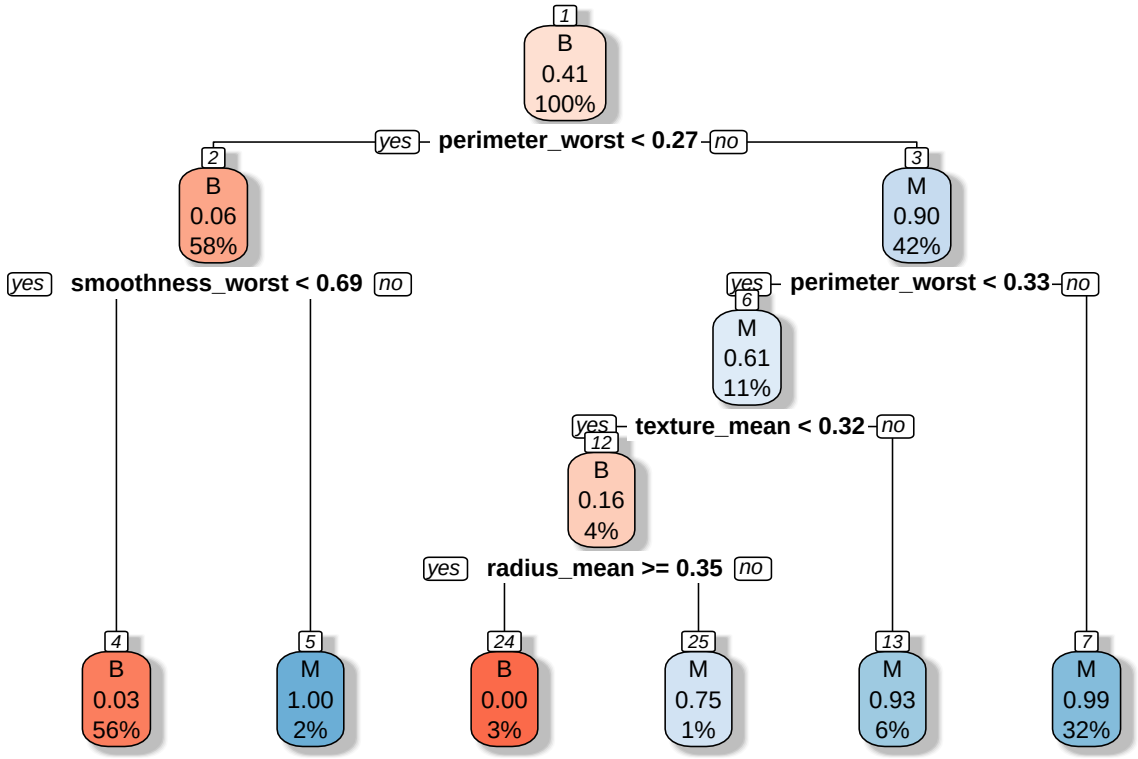
```
## The following object is masked from 'package:varImp':
##
##     varImp
```

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

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

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

```
best_decision_tree <- rpart(as.factor(wdbc_train_labels)~., data = wdbc_train,
                            control = rpart.control(minsplit = 11,
                                                    maxdepth = 10))
rpart.plot(x = best_decision_tree, box.palette="RdBu", shadow.col="gray", nn=TRUE, yesno = 2)
```



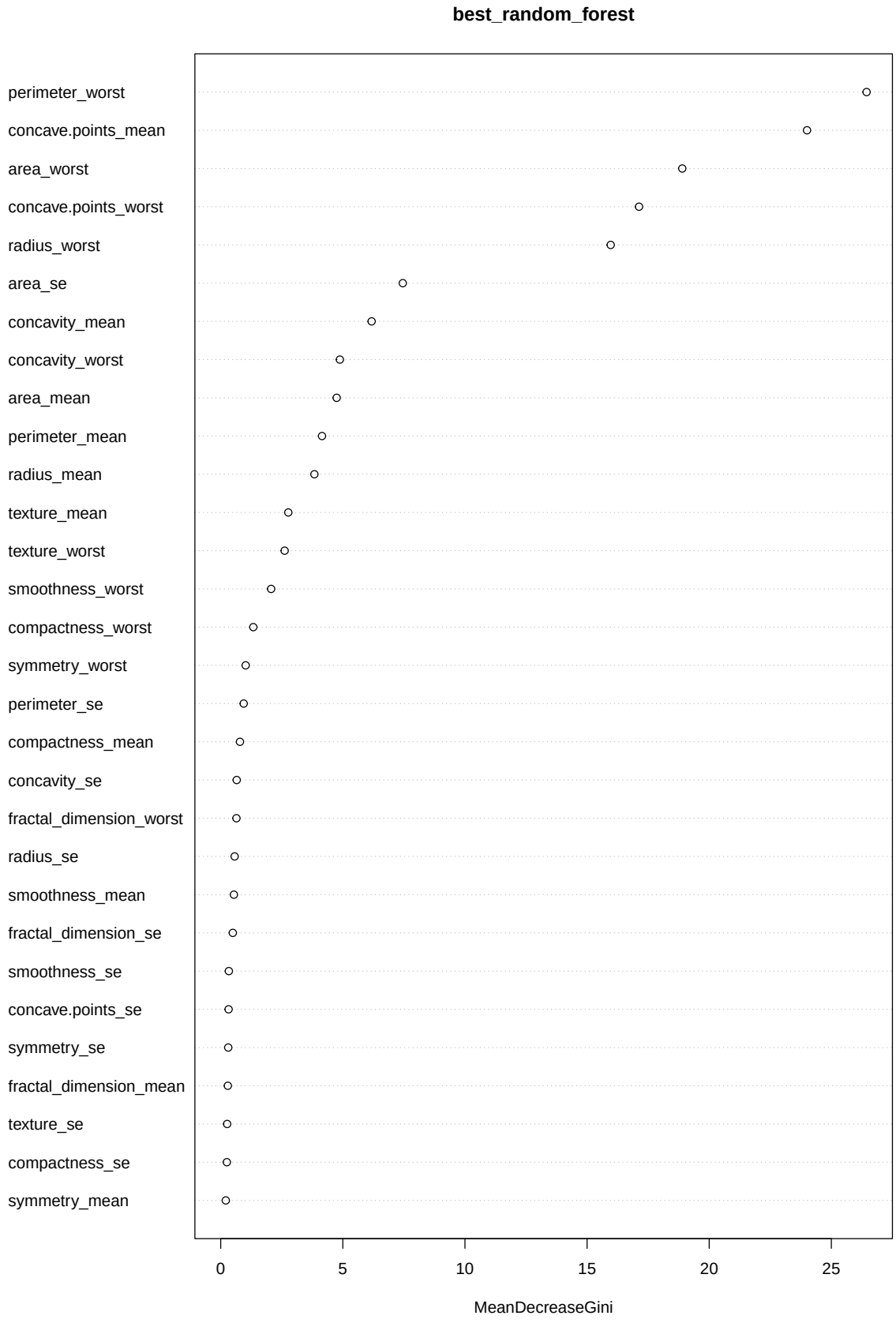
۳.۴ الگوریتم جنگل تصادفی

اندازه ها با جستجوی شبکه ای مشخص شده اند.

```
best_random_forest <- randomForest(as.factor(wdbc_train_labels)~ ., data = wdbc_train,
                                   nodesize = 9,
                                   sampsize = 329,
                                   mtry = 7,
                                   ntree = 210)
best_random_forest
```

```
##
## Call:
##  randomForest(formula = as.factor(wdbc_train_labels) ~ ., data = wdbc_train,      nodesize = 9, samps
##                Type of random forest: classification
##                      Number of trees: 210
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 4.66%
## Confusion matrix:
##     B   M class.error
## B 242  10  0.03968254
## M  10 167  0.05649718
```

```
# Identify the most significant independent variables
varImpPlot(best_random_forest)
```


best_random_forest
perimeter_worst
concave.points_mean
area_worst
concave.points_worst
radius_worst
area_se
concavity_mean
concavity_worst
area_mean
perimeter_mean
radius_mean
texture_mean
texture_worst
smoothness_worst
compactness_worst
symmetry_worst
perimeter_se
compactness_mean
concavity_se
fractal_dimension_worst
radius_se
smoothness_mean
fractal_dimension_se
smoothness_se
concave.points_se
symmetry_se
fractal_dimension_mean
texture_se
compactness_se
symmetry_mean
0
5
10
15
20
25
MeanDecreaseGini

۴.۴ بردار ماشین های پشتیبان

۱.۴.۴ مدل با هسته های خطی ، شعاعی و چند جمله ای

```
library(tidyverse)
library(dplyr)
library(caret)
library(corrr)
```

```
## Warning: package 'corrr' was built under R version 4.1.2
```

```
library(DT)
```

```
## Warning: package 'DT' was built under R version 4.1.2
```

```
library(e1071)
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:modeltools':
##
##     prior
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
wdbc1=read.csv("C:/Users/atusa/Downloads/data.csv",header=TRUE)
wdbc1=wdbc1[,-33]
wdbc1=wdbc1[,-1]
wdbc_train1 <- wdbc1[1:429,]
wdbc_test1 <- wdbc1[430:569,]
###### linear kernel #############
cost_range <-c(0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2, 5)
tune.out <- tune(svm, as.factor(diagnosis)~., data = wdbc_train1, kernel = "linear",
                 ranges = list(cost=cost_range))

bestmod_linear <- tune.out$best.model
summary(bestmod_linear)
```

```
##
## Call:
## best.tune(method = svm, train.x = as.factor(diagnosis) ~ ., data = wdbc_train1,
##     ranges = list(cost = cost_range), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  101
```

```
##
##  ( 50 51 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
```

```
####### confusion matrix for linear ##########
predictions_train <- predict(bestmod_linear)
confusionMatrix(predictions_train, as.factor(wdbc_train1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 251   6
##          M   1 171
##
##                Accuracy : 0.9837
##                  95% CI : (0.9667, 0.9934)
##     No Information Rate : 0.5874
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9662
##
##  Mcnemar's Test P-Value : 0.1306
##
##             Sensitivity : 0.9960
##             Specificity : 0.9661
##          Pos Pred Value : 0.9767
##          Neg Pred Value : 0.9942
##              Prevalence : 0.5874
##          Detection Rate : 0.5851
##    Detection Prevalence : 0.5991
##       Balanced Accuracy : 0.9811
##
##        'Positive' Class : B
##
```

```
############## accuracy for test data ##################
predictions_test1 <- predict(bestmod_linear, newdata = wdbc_test1)
confusionMatrix(predictions_test1, as.factor(wdbc_test1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 104   1
##          M   1  34
##
##                Accuracy : 0.9857
##                  95% CI : (0.9493, 0.9983)
##     No Information Rate : 0.75
```

```
##     P-Value [Acc > NIR] : 3.641e-15
##
##                   Kappa : 0.9619
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9905
##             Specificity : 0.9714
##          Pos Pred Value : 0.9905
##          Neg Pred Value : 0.9714
##              Prevalence : 0.7500
##          Detection Rate : 0.7429
##    Detection Prevalence : 0.7500
##       Balanced Accuracy : 0.9810
##
##        'Positive' Class : B
##
```

```
############ Polynomial kernel ################
tune.out2 <- tune(svm,  as.factor(diagnosis)~., data = wdbc_train1, kernel = "polynomial",
                  ranges = list(cost = cost_range))

bestmod_polynomial <- tune.out2$best.model
summary(bestmod_polynomial)
```

```
##
## Call:
## best.tune(method = svm, train.x = as.factor(diagnosis) ~ ., data = wdbc_train1,
##     ranges = list(cost = cost_range), kernel = "polynomial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  5
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  110
##
##  ( 54 56 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
```

```
######## confusion matrix for Polynomial ##########
predictions_train2 <- predict(bestmod_polynomial)
confusionMatrix(predictions_train2, as.factor(wdbc_train1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction   B   M
##          B 252  13
##          M   0 164
##
##                Accuracy : 0.9697
##                  95% CI : (0.9487, 0.9838)
##     No Information Rate : 0.5874
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9368
##
##  Mcnemar's Test P-Value : 0.0008741
##
##             Sensitivity : 1.0000
##             Specificity : 0.9266
##          Pos Pred Value : 0.9509
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5874
##          Detection Rate : 0.5874
##    Detection Prevalence : 0.6177
##       Balanced Accuracy : 0.9633
##
##        'Positive' Class : B
##
```

```
############## accuracy for test data ##################
predictions_test2 <- predict(bestmod_polynomial, newdata = wdbc_test1)
confusionMatrix(predictions_test2, as.factor(wdbc_test1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 105   2
##          M   0  33
##
##                Accuracy : 0.9857
##                  95% CI : (0.9493, 0.9983)
##     No Information Rate : 0.75
##     P-Value [Acc > NIR] : 3.641e-15
##
##                   Kappa : 0.9612
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.9429
##          Pos Pred Value : 0.9813
##          Neg Pred Value : 1.0000
##              Prevalence : 0.7500
##          Detection Rate : 0.7500
##    Detection Prevalence : 0.7643
##       Balanced Accuracy : 0.9714
##
##        'Positive' Class : B
```

```
##
```

```
################ Radial kernel ####################
gamma_range = c(0.5,1,2,3,4)

tune.out23 <- tune(svm, as.factor(diagnosis) ~., data=wdbc_train1 , kernel = "radial",
                ranges = list(cost = cost_range,
                              gamma = gamma_range))
bestmod_radial <- tune.out23$best.model
summary(bestmod_radial)
```

```
##
## Call:
## best.tune(method = svm, train.x = as.factor(diagnosis) ~ ., data = wdbc_train1,
##     ranges = list(cost = cost_range, gamma = gamma_range), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  401
##
##  ( 176 225 )
##
##
## Number of Classes:  2
##
## Levels:
##  B M
```

```
######## confusion matrix for radial ##########

predictions_train3 <- predict(bestmod_radial)
confusionMatrix(predictions_train3, as.factor(wdbc_train1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   B   M
##          B 252   0
##          M   0 177
##
##                Accuracy : 1
##                  95% CI : (0.9914, 1)
##     No Information Rate : 0.5874
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
```

```
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5874
##          Detection Rate : 0.5874
##    Detection Prevalence : 0.5874
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : B
##
```

```
############# accuracy for test data ##################
predictions_test <- predict(bestmod_radial, newdata =wdbc_test1)
confusionMatrix(predictions_test, as.factor(wdbc_test1$diagnosis))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  B  M
##          B 78  0
##          M 27 35
##
##                Accuracy : 0.8071
##                  95% CI : (0.7319, 0.8689)
##     No Information Rate : 0.75
##     P-Value [Acc > NIR] : 0.06872
##
##                   Kappa : 0.5909
##
##  Mcnemar's Test P-Value : 5.624e-07
##
##             Sensitivity : 0.7429
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.5645
##              Prevalence : 0.7500
##          Detection Rate : 0.5571
##    Detection Prevalence : 0.5571
##       Balanced Accuracy : 0.8714
##
##        'Positive' Class : B
##
```

برای تعیین مقدار بهینه c درهر مدل از روش 10-fold cross validation استفاده کردیم تا کمترین مقدار خطا در هر مدل را پیدا کنیم به علاوه دقت مدل بهینه را نیز برای داده های آموزشی و آزمایشی براورد کرده ایم.

۱. برای مدل خطی بهترین مدل با c=0.05 برازش داده شده است. دقت مدل با داده های آموزشی برابر ۹۷ درصد و با داده های آزمایشی ۹۸ درصد می باشد.

۲. برای مدل چند جمله ای مقدار بهینه c=5 است و درجه معادله ۳ براورد شده است.دقت مدل با داده های آموزشی ۹۶ درصد و با داده های آزمایشی ۹۸ درصد است.

۳. برای مدل شعاعی دقت مدل با داده های آموزشی به ۱۰۰ درصد می رسد اما در داده های آزمایشی دقت برابر ۸۰ درصد است.

۵ نتیجه گیری

در این پروژه مجموعه داده های سرطان سینه ویسکانیسین را مورد بررسی قرار دادیم و در نهایت به این دقت براود برای مد های برازش داده شده رسیدیم:

جدول ۲: Table-2

مدل	دقت براورد شده برای داده های آموزشی و آزمایشی
SVM whit Linear kernel	train=0.97 , test=0.98
SVM whit Polynomial kernel	train=0.96 , test=0.98
SVM whit Radial kernel	train=1 , test=0.80
knn	train=0.96 , test=0.98
random forest	train=0.94 , test=lower accuracy

بهترین مدل در این گزارش براساس بردار ماشین های پشتیبان با هسته چند جمله ای شناخته میشود.