Breast Cancer Detection

Wilsven Leong

8/21/2021

Introduction

According to the American Cancer Society, breast cancer is the most common cancer in American women, except for skin cancers. The average risk of a woman in the United States developing breast cancer sometime in her life is about 13%. This means there is a 1 in 8 chance she will develop breast cancer.

Trends in Breast Cancer Incidence

Incidence rates have increased by 0.5% annually in recent years.

Trends in Breast Cancer Deaths

Breast cancer is the second leading cause of cancer death in women and the chance that a woman will die from breast cancer is about 1 in 39 (about 2.6%). Since 2007, breast cancer death rates have been steady in women younger than 50, but have continued to decrease in older women. From 2013 to 2018, the death rate went down by 1% per year.

One of the reasons for these decreases is believed to be the result of finding breast cancer earlier through screening which will be the focus of this project.

Objective

In this project, classification models will aim to determine whether a tumor is benign and malignant by identifying cytological attributes (features) which are significant in breast cancer patients. This project will make use of data from a study on breast cancerreferring to 699 patients. The actual data can be found at UCI Machine Learning Repository. The variables were computed from a digitized image of a breast mass and describe characteristics of the cell nucleus present in the image. In particular the features are tabulated in the following:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values) c) perimeter d) area e) smoothness (local variation in radius lengths) f) compactness (perimeterˆ2 / area - 1.0) g) concavity (severity of concave portions of the contour) h) concave points (number of concave portions of the contour) i) symmetry j) fractal dimension (“coastline approximation” - 1)

Data Cleaning

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id_number              : 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 ...
```

We should start by checking for missing values.

```
## [1] 0
```

We will also remove the `id_number` variable which doesn't provide value to our classification models. We will also convert our `diagnosis` varaible from character into factor.

```
##  diagnosis  radius_mean      texture_mean   perimeter_mean     area_mean
##  B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5
##  M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3
##            Median :13.370   Median :18.84   Median : 86.24   Median : 551.1
##            Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9
##            3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7
##            Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0
##  smoothness_mean   compactness_mean  concavity_mean    concave_points_mean
##  Min.   :0.05263   Min.   :0.01938   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031
```

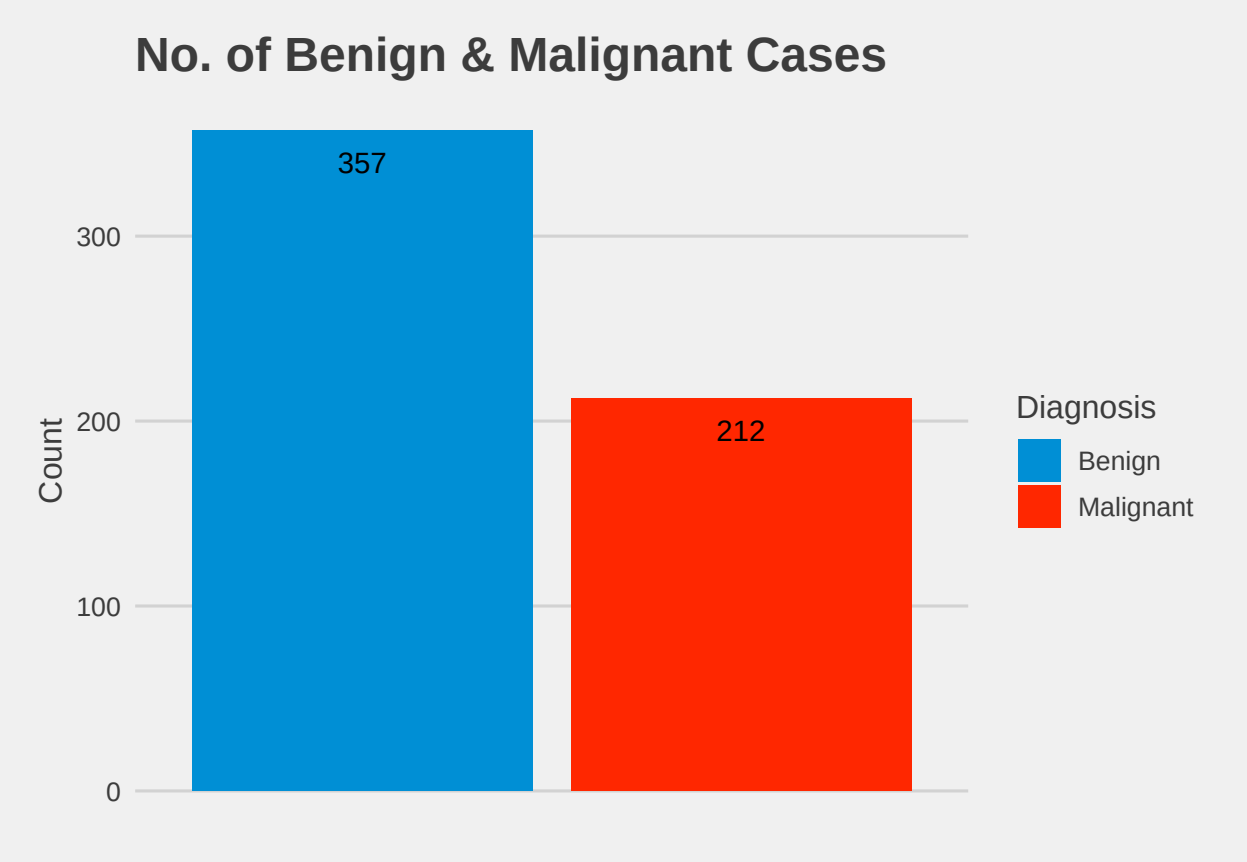
```
##  Median :0.09587   Median :0.09263   Median :0.06154   Median :0.03350
##  Mean   :0.09636   Mean   :0.10434   Mean   :0.08880   Mean   :0.04892
##  3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400
##  Max.   :0.16340   Max.   :0.34540   Max.   :0.42680   Max.   :0.20120
##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

Exploratory Data Analysis

Looking at the proportions of *benign* and *malignant* observations, we are fortunate that this data set does not suffer from *class imbalance*. Class imbalance refers to when a target class within a data set is outnumbered by the other target class (or classes). This can lead to misleading accuracy metrics, known as accuracy paradox. High acurracies can be obtained even when making predictions simply by guessing.

Diagnosis	Proportions
B	0.6274165
M	0.3725835

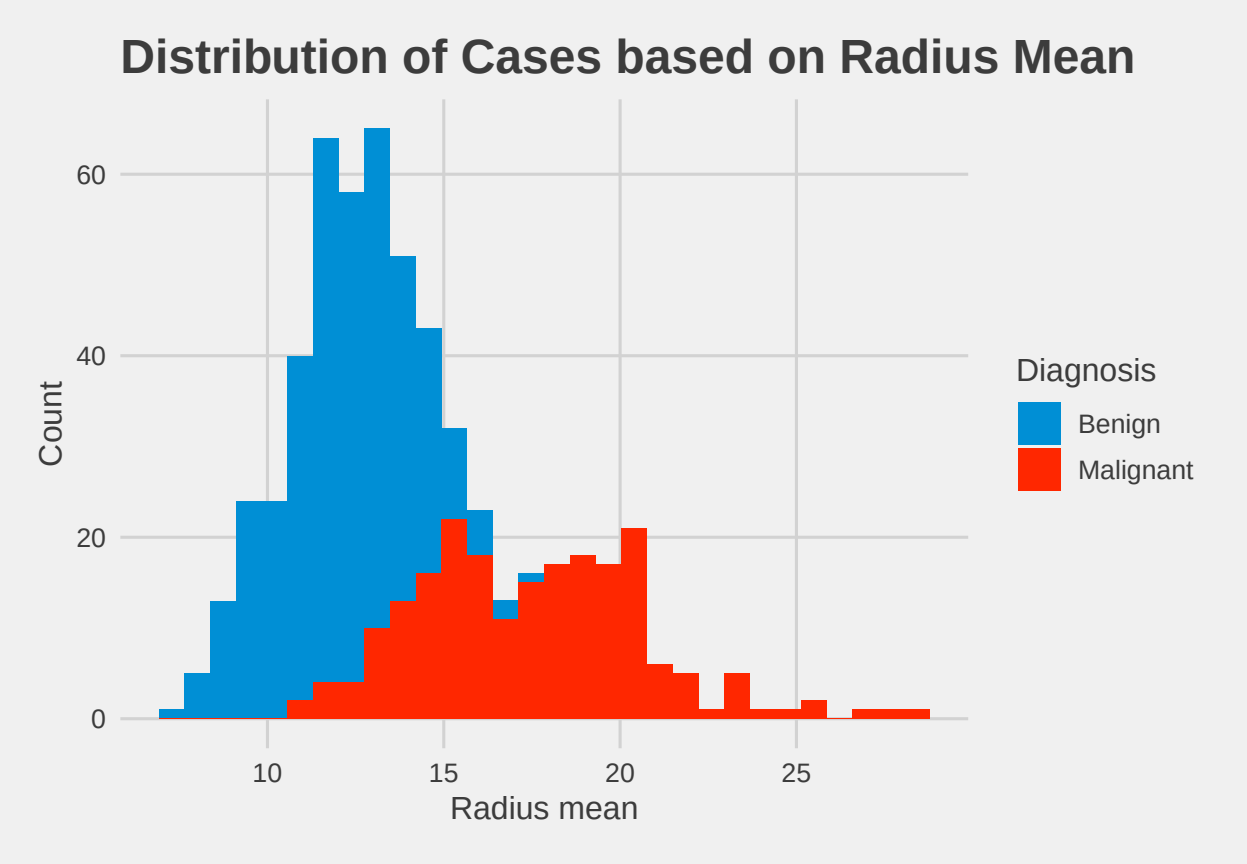
We can also visualize the number of diagnosis results in a countplot.

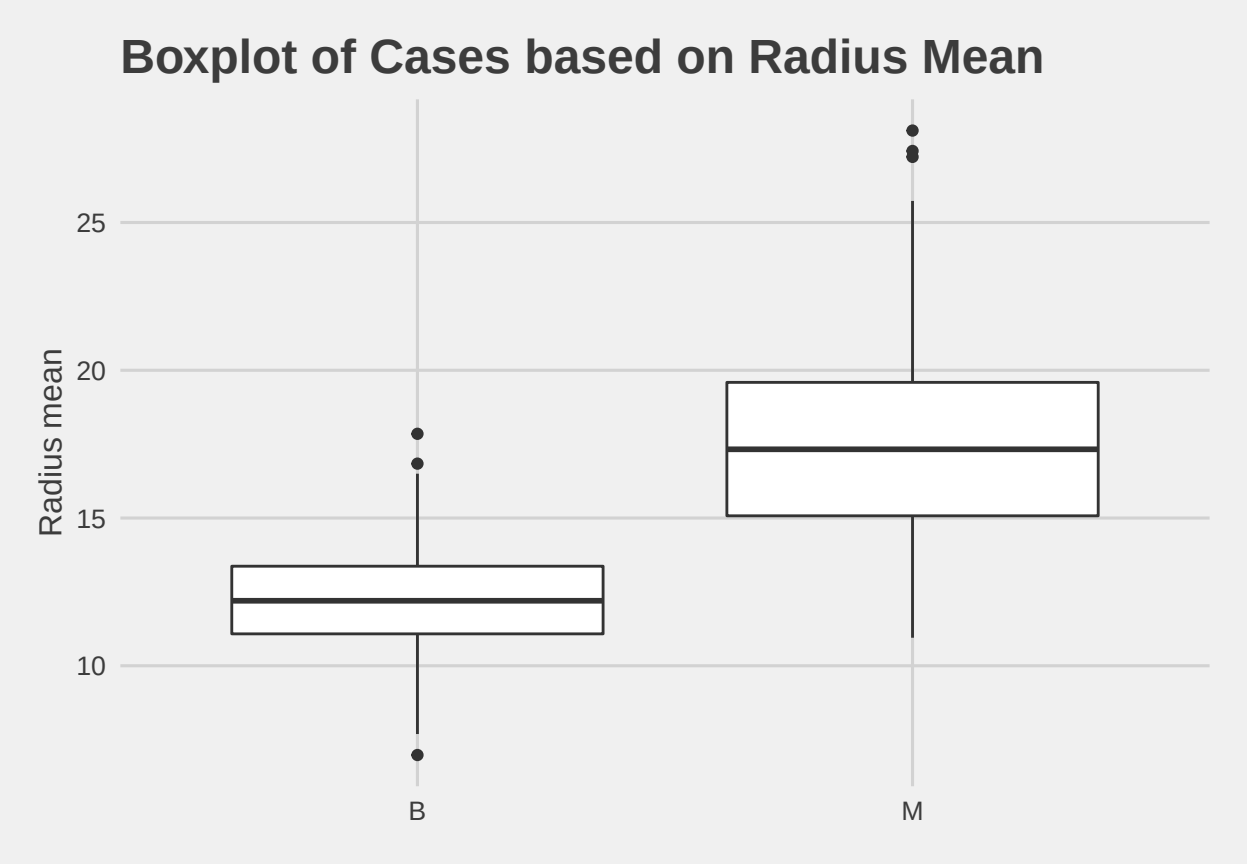


After data cleaning, we can see that we now have 569 valid observations, of which 357 has a *benign* breast tumor and the other 212 has a *malignant* breast tumor.

Univariate Data Analysis

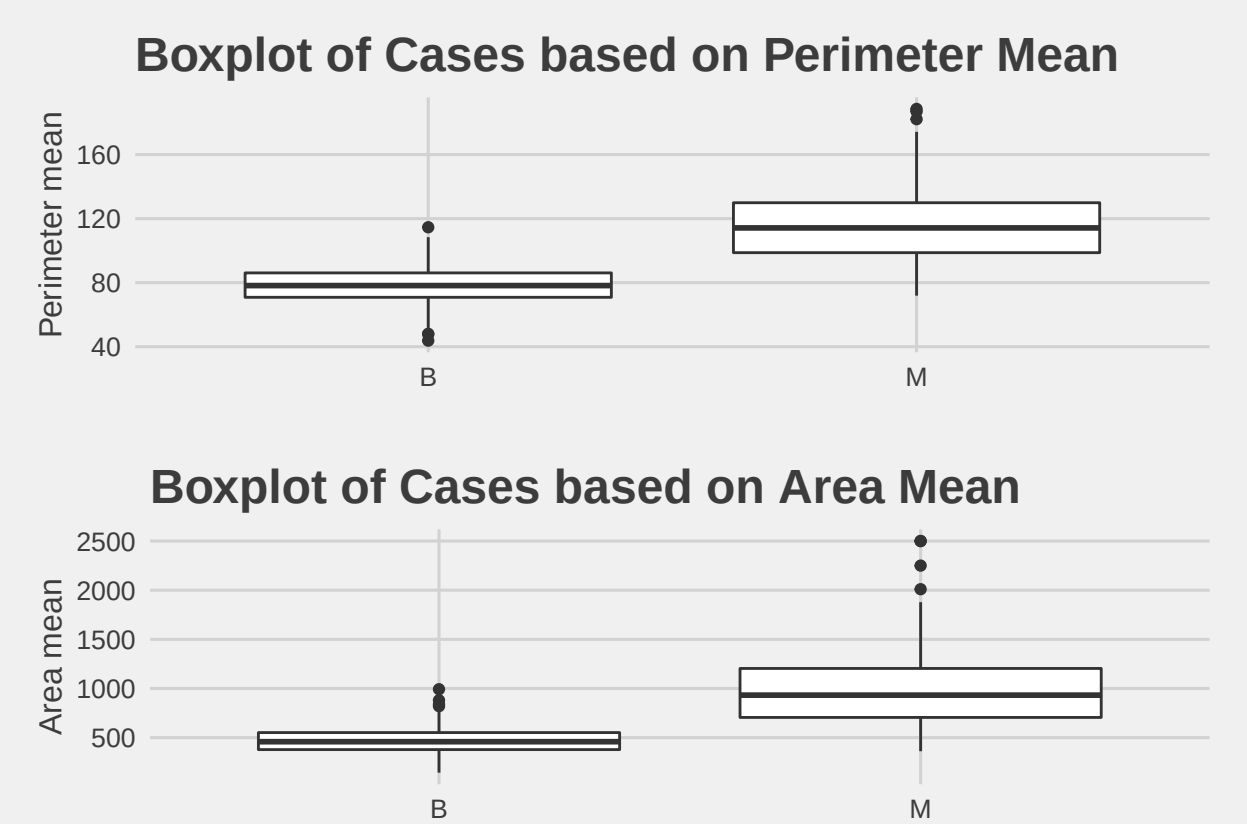
We will perform some univariate data analysis to get an idea how cases might be dependent on the variables, if any. Let's take a look at `diagnosis` and `radius_mean`.


Distribution of Cases based on Radius Mean
Count
60
40
20
0
10
15
20
25
Radius mean
Diagnosis
Benign
Malignant



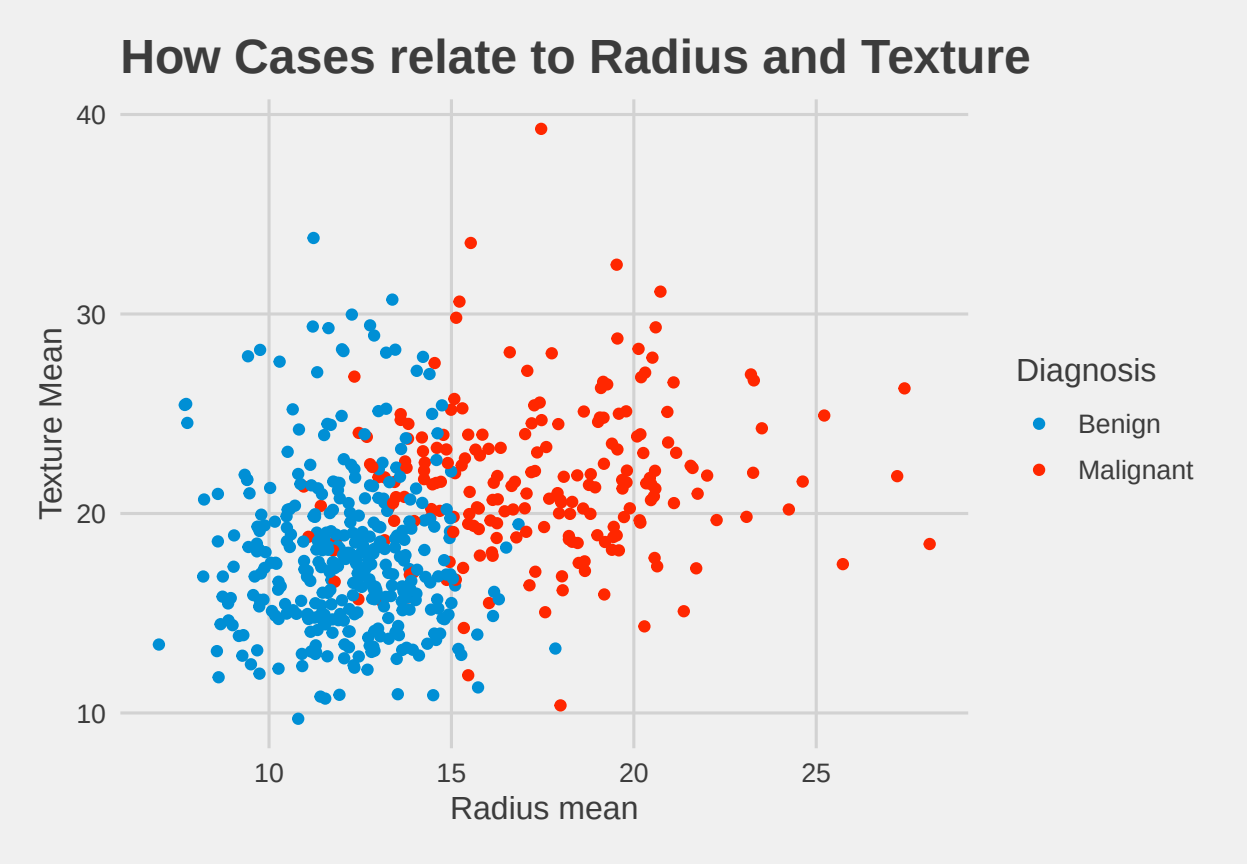
This univariate analysis shows that large tumor radius most likely belongs to *malignant* tumors. It should already make sense that when radius is large, perimeter and area of the tumor which follow a linear relationship with radius, will also be large.

Therefore, I will only be showing the boxplots of perimeter and area in the following visualizations to further showcase my point.



Bivariate Data Analysis

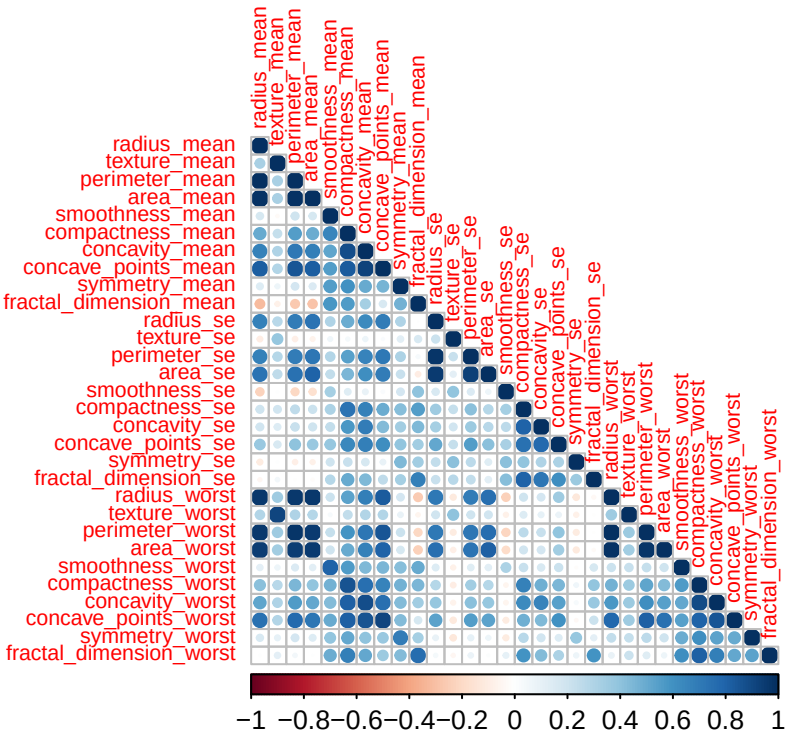
It would also be interesting to investigate how some independent variables relate to one another and how the `diagnosis` depend on said relationships.



We can see that with two variables only, we can visually separate most of the *benign* and *malignant* tumors. Obviously, this isn't sufficient and we will have to make use of more independent variables in our classification models to accurately predict as many correct cases as possible.

Correlation Plot

Let"s now view the correlation plot between the independent variables.



Looking at the correlation between the independent variables, we can see that some variables are highly correlated. This can potentially lead to problems arising from multicollinearity.

Recall in our univariate and bivariate analyses above that `radius_mean`, `perimeter_mean` and `area_mean` are highly correlated with one another.

```
## Compare row 7  and column  8 with corr  0.921
##   Means:  0.571 vs 0.389 so flagging column 7
## Compare row 8  and column  28 with corr  0.91
##   Means:  0.542 vs 0.377 so flagging column 8
## Compare row 23  and column  21 with corr  0.994
##   Means:  0.48 vs 0.367 so flagging column 23
## Compare row 21  and column  3 with corr  0.969
##   Means:  0.446 vs 0.359 so flagging column 21
## Compare row 3  and column  24 with corr  0.942
##   Means:  0.414 vs 0.353 so flagging column 3
## Compare row 24  and column  1 with corr  0.941
##   Means:  0.39 vs 0.349 so flagging column 24
## Compare row 1  and column  4 with corr  0.987
##   Means:  0.35 vs 0.347 so flagging column 1
## Compare row 13  and column  11 with corr  0.973
##   Means:  0.372 vs 0.346 so flagging column 13
## Compare row 11  and column  14 with corr  0.952
##   Means:  0.323 vs 0.347 so flagging column 14
## Compare row 22  and column  2 with corr  0.912
##   Means:  0.224 vs 0.357 so flagging column 2
## All correlations <= 0.9
```

Diagnosis	Proportion
B	0.627193
M	0.372807

Diagnosis	Proportion
B	0.6283186
M	0.3716814

```
##  [1] "compactness_mean"     "concavity_mean"       "texture_worst"
##  [4] "fractal_dimension_se" "texture_mean"         "perimeter_worst"
##  [7] "diagnosis"            "texture_se"           "perimeter_se"
## [10] "radius_mean"
```

There are ten features with correlation higher than 0.9. Excluding these variables from unsupervised machine learning algorithms when developing for predictive models may be beneficial.

However, since the models we will be building involves supervised machine learning algorithms, we will leave all variables untouched.

We split the data set into our training and test sets in a 80-20% split. We will use the training set to train our model along with some optimization of the hyperparameters, and use our test set as the unseen data. This will be a useful final metric to let us know how well our model does.

Splitting the Data Set

As we can see, the proportion of cases in both the train and test sets are similar.

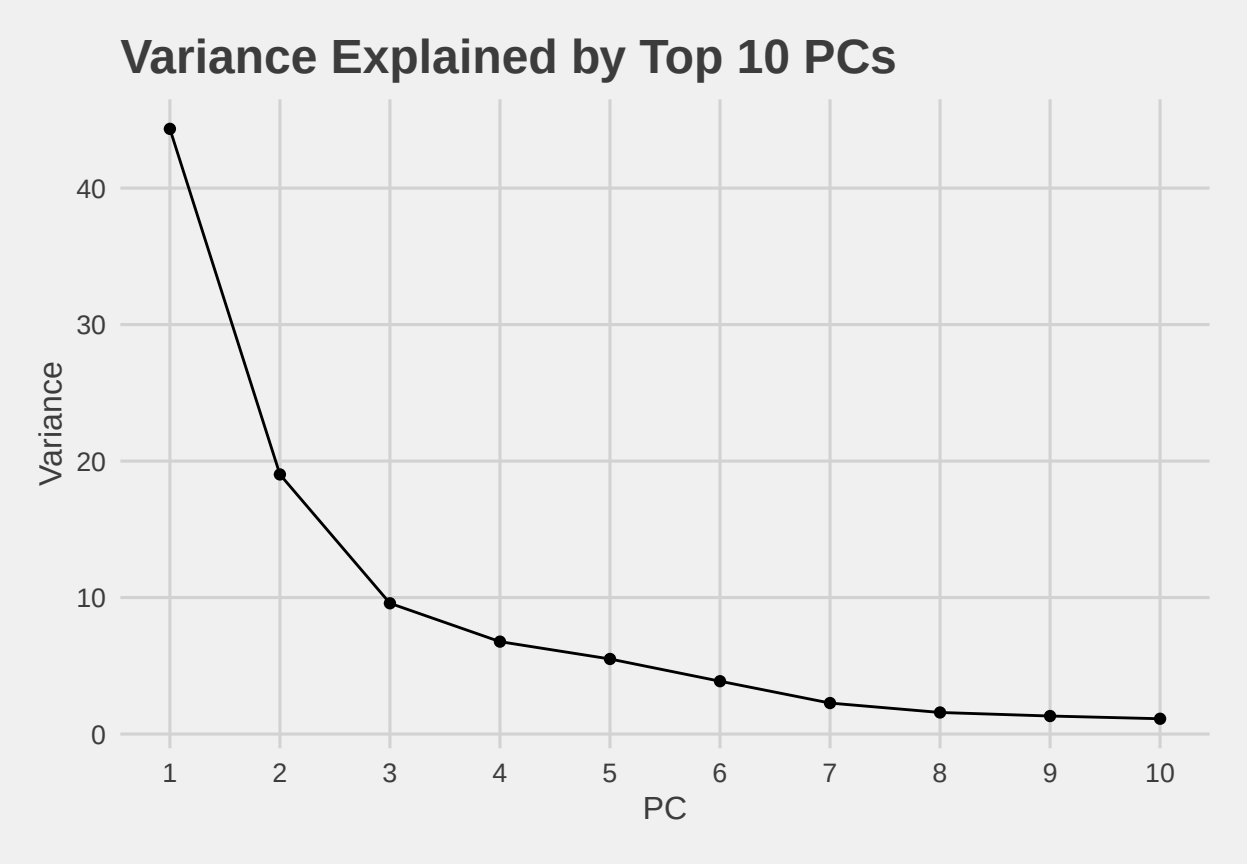
Principal Component Analysis

Principal component analysis (PCA) is a technique for transforming data sets in order to reduce dimensionality without reducing the number of features. This is done by identifying the principal components which explain as much of the data variance as possible. PCA can be used to improve visualization of multidimensional data and, potentially, to improve the predictive accuracy of classification models.

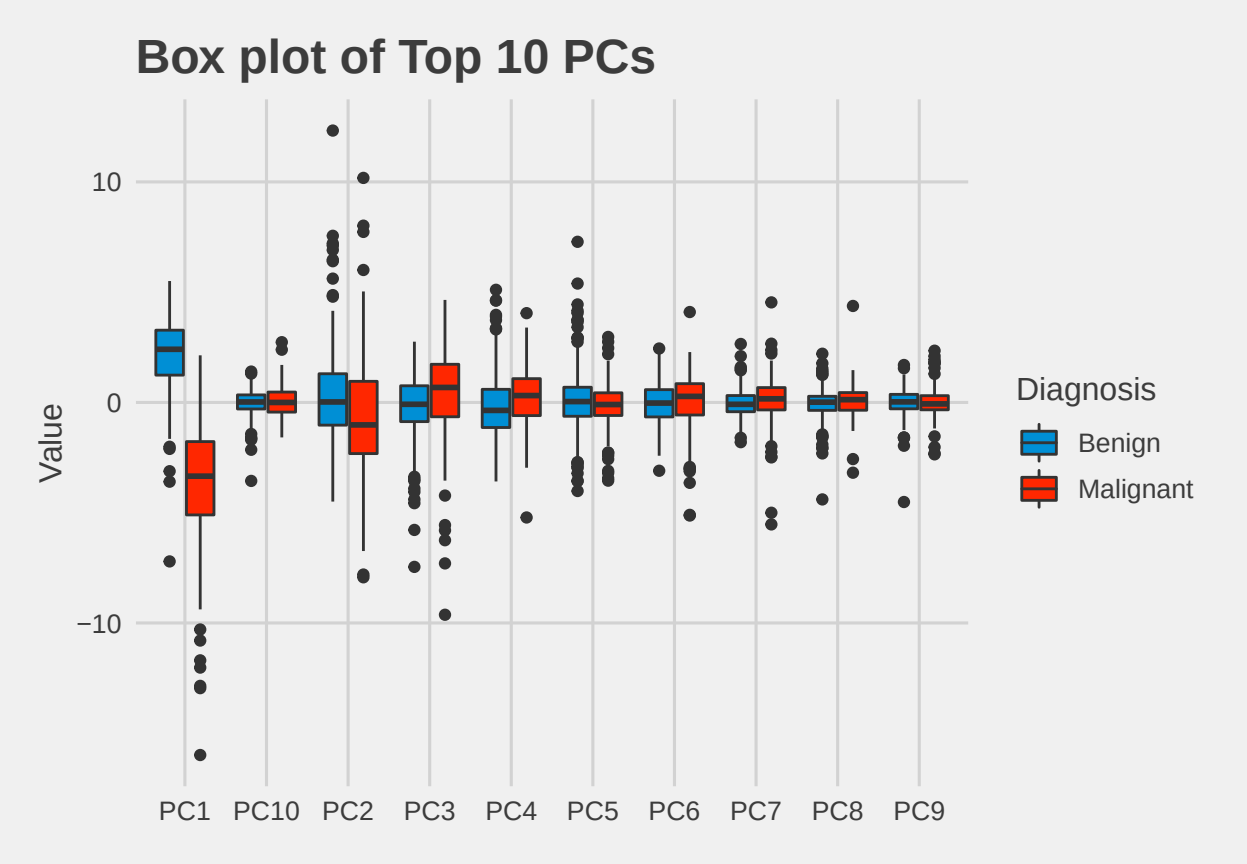
Below is a table and plot of the percentages of variance explained by the top 10 principal components.

Table 1: Variance Explained by Top 10 PCs

PC	Variance
1	44.34
2	19.02
3	9.58
4	6.77
5	5.50
6	3.87
7	2.27
8	1.58
9	1.32
10	1.12

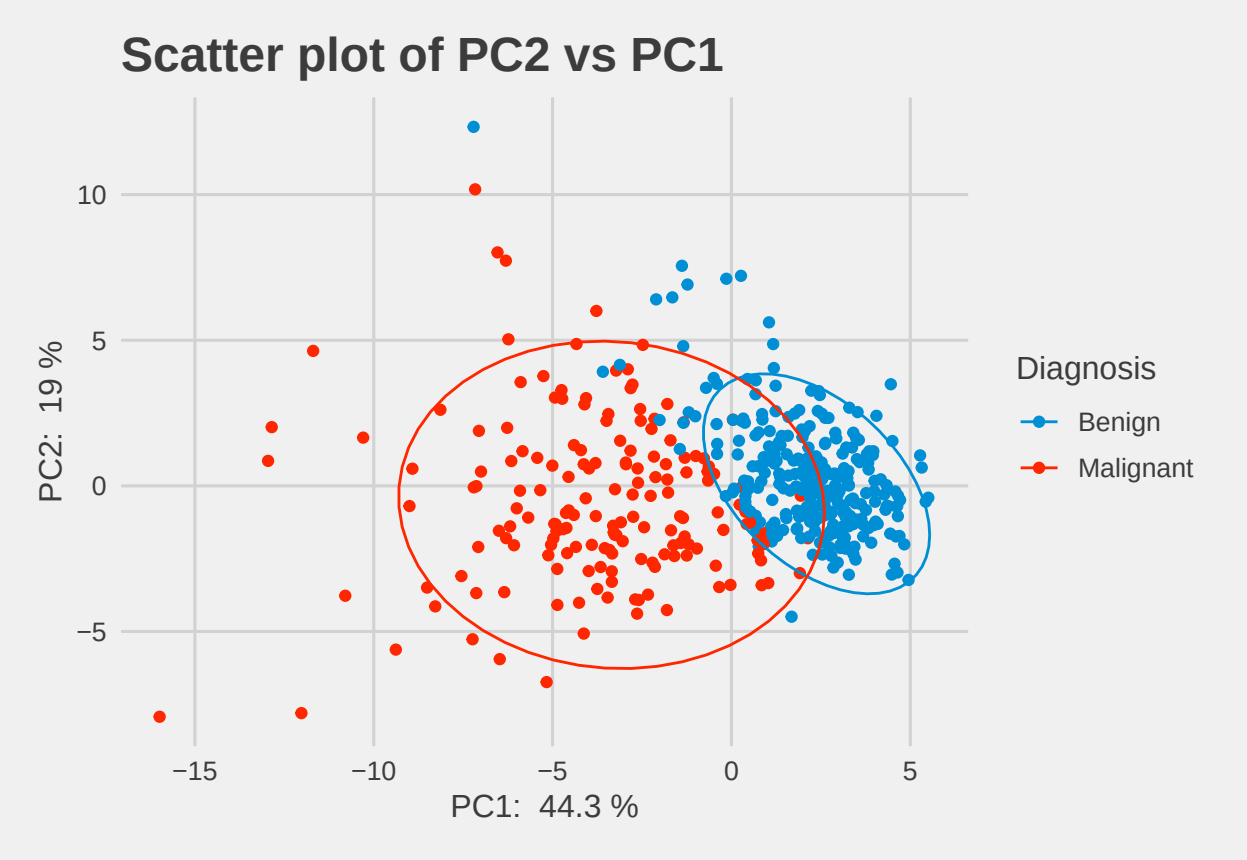


Let's also plot box plots for each of the first 10 principal components grouped by diagnosis. In most cases the spread is greater for *malignant* tumors than for *benign* tumors. PC1 is the only component for which the interquartile ranges do not overlap. Principal component analysis does not take into account the classification of data, in this case the diagnosis assigned to each sample.



Below is the two-dimensional scatter plot of the first two principal components. From the plot, it shows that the *malignant* data points are more spreaded out than the *benign* data points and that more of the variance can be accounted for on the x-axis (PC1) than on the y-axis (PC2).

The two ellipses drawn on the plot help to visualize this even better. A larger ellipse is needed for the *malignant* data points than for *benign* data points. A distinct separation of data by classification visually is possible, despite some overlap. Therefore, this analysis support the use of PCA in classification algorithm development to predict diagnosis from this data set.



Classification Models

Classification models aim to predict the target class for new observations, that is, predicting the output from a given set of predicting or independent variables. In this project, we will train several classification models including Naive Bayes, Logistic Regression, Decision Tree and lastly, Random Forest.

Naive Bayes

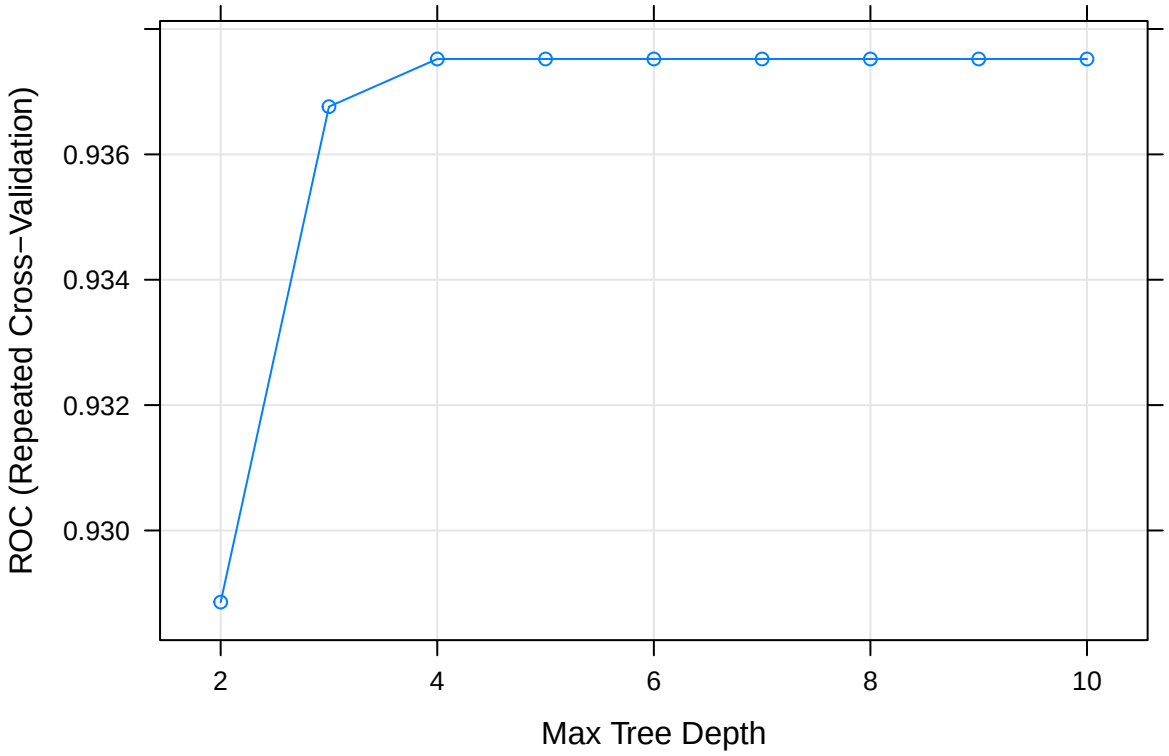
The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large data sets. Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

Logistic Regression

Logistic regression is probably the most commonly used form of generalized linear model (GLM). Linear regression assumes that the predictor, X, and the outcome Y, follow a bivariate normal distribution such that the conditional expectation, i.e. the expected outcome Y for a given predictor X, fits the regression line. Logistic regression is therefore an extension of linear regression.

Logistic Regression (PCA)

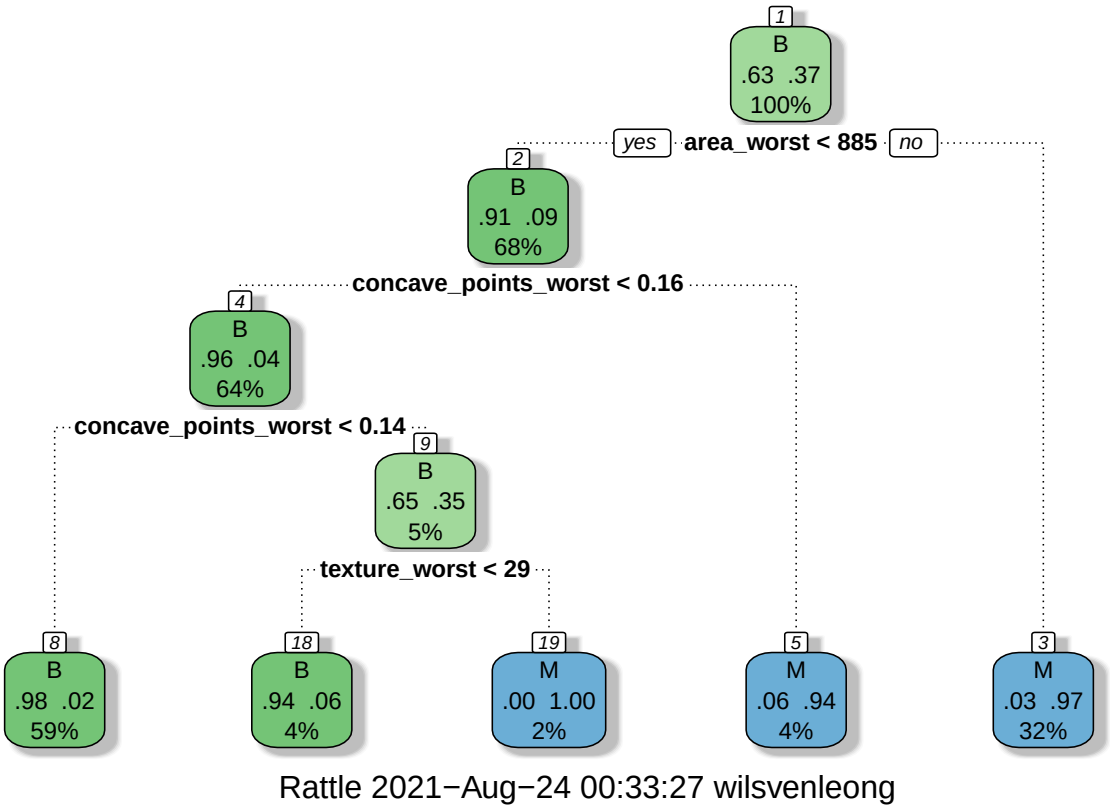
Decision Tree



From the ROC plot above, the optimal max tree depth is 4.

Table 2: Final Results

	Accuracy	Senstivity	Specificity	F1	False.Neg..Rate	False.Pos..Rate
Naive Bayes	0.91	0.90	0.92	0.88	0.10	0.08
Logistic Regression	0.95	0.90	0.97	0.93	0.10	0.03
Logistic Regression (PCA)	0.96	0.95	0.96	0.94	0.05	0.04
Decision Tree	0.93	0.90	0.94	0.90	0.10	0.06
Random Forest	**0.97**	**0.95**	**0.99**	**0.96**	**0.05**	**0.01**



Random Forest

As previously described, models can suffer from diminished performance due to multidimensionality of data. PCA can be useful to reduce problems with multicollinearity by reducing the number of features required for pre-processing. Decisions trees are another way to address this issue, effectively partitioning the data such that final predictions can be made on a smaller subset of predictors. This is also known as "Bagging".

Bagging, also known as bootstrap aggregation, helps avoid overfitting to the training set by effectively creating an ensemble ('forest') of multiple decision trees and averaging over all the predictions from each of these trees to form a final prediction.

Conclusion

From our initial exploration of the data, we have hypothesized that tumor size is a significant predictor of whether a tumor is benign or malignant. In our final conclusion, we can see that all the models performed extremely well and this can be further explained by the extreme difference in size features (i.e. radius, perimeter and area) between benign and malignant tumors.

In our case of cancer prediction, it is crucial that we minimize our Type II error. Since a *malignant* diagnosis is regarded as a *positive* test, we should be aiming to maximize our Sensitivity of our model.

From the table above, we can see that Logistic Regression with PCA and Random Forest have the highest Sensitivity. While Type I error is less expensive than Type II, it is nonetheless desirable and in this case, a perfect tiebreaker for our chosen model.

Once again, from the table above, we can see that the Random Forest takes the cake with a whooping Specificity of 99.0%, beating all other models.