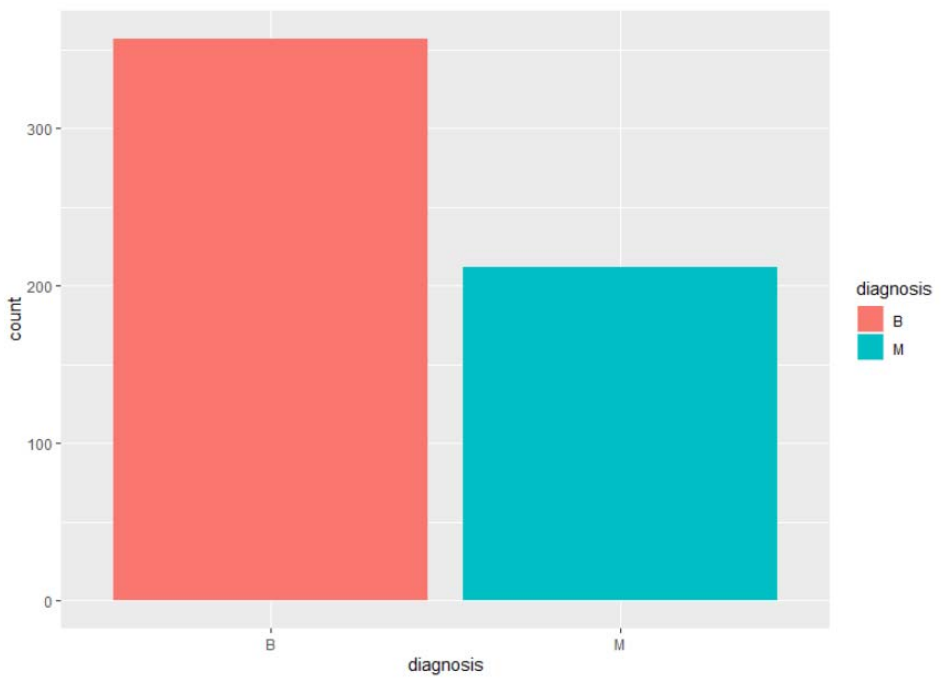
Problem Statement

Breast cancer is the most common type of cancer for women. Luckily, breast cancer death rates have declined 40% from 1989 to 2016. The progress is attributed to improvements in early detection. By being able to detect all the people who have breast cancer, the death rate can be lowered even more. This is serious problem because you are dealing with matters of life and death. A classification method would be a great option for this because the data already contains the labels or malignant or benign for the tumor biopsies and you want to be able to tell people if they have or do not have breast cancer.

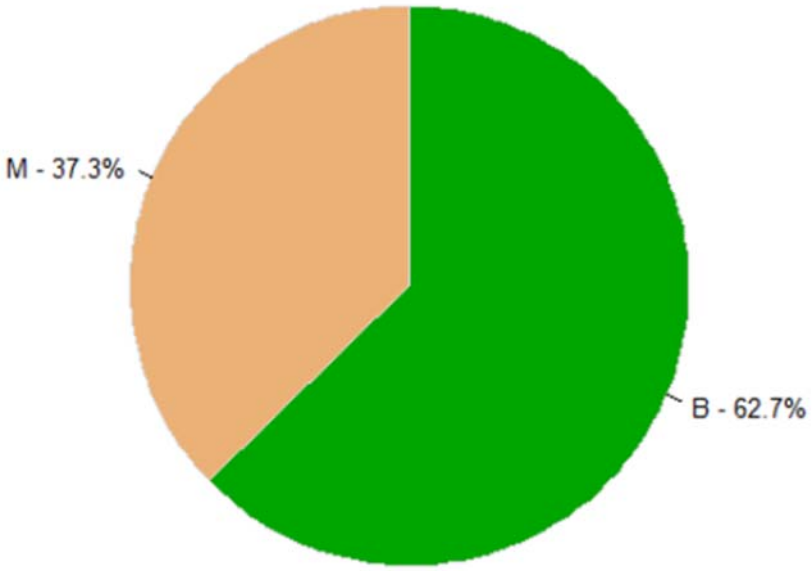
Data

I used the breast cancer Wisconsin (Diagnostic) data set that is available on kaggle. (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) It was collected by the University of Wisconsin General Surgery Department. There was an academic paper written about the data set titled: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets". I am sure the data was collected in order to figure out a method to determine if someone had a malignant (bad result) or benign (good result) form of a breast tumor from a biopsy. There are 32 columns and 569 rows of data. This data probably only came from one hospital in Madison, WI. This means that the results might not be applicable outside of Wisconsin, Midwest, or the USA. The first column in the data set is the subject's id. The second column is the diagnosis (answer/label) of whether the tumor is malignant or benign. The other 30 columns are made up of three variations of 10 variables: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The variables are computed from a digitized image of a fine needle aspirate (FNA) biopsy of a breast mass. The three variations of those 10 variables are: mean, se (standard error), and worst. Mean is the average value of the measure detected by the computer, se is the standard error of variable being detected, and worst is the average of the three largest values detected. Most of the values in each variables are in in decimal format meaning less than one or up to around a thousand except for the id variables which has up to 9 digits present. The id variable will not be useful in helping with our classification procedure so it was eliminated. Below is the summary of the data.

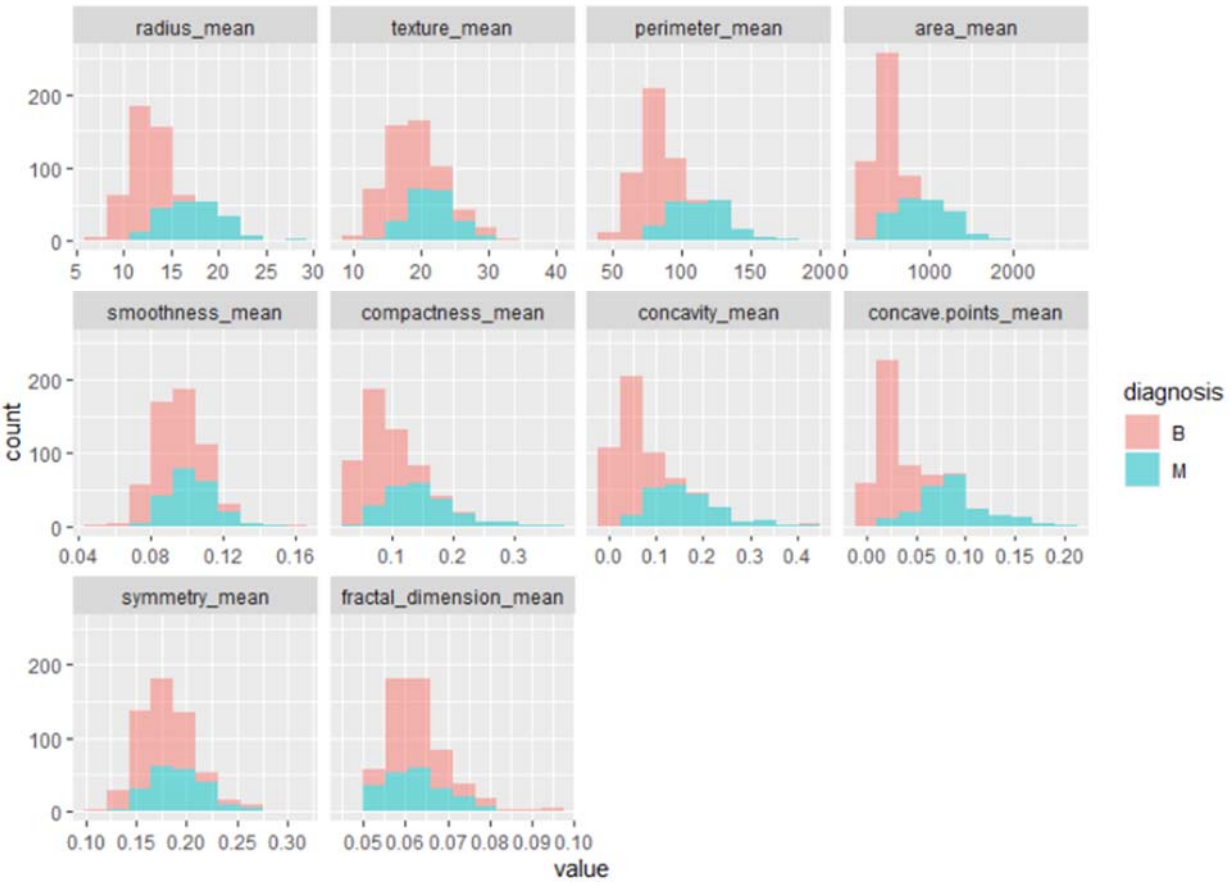
```
> summary(Data)
 diagnosis  radius_mean      texture_mean   perimeter_mean     area_mean      smoothness_mean
 B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5   Min.   :0.05263
 M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637
           Median :13.370   Median :18.84   Median : 86.24   Median : 551.1   Median :0.09587
           Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9   Mean   :0.09636
           3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530
           Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0   Max.   :0.16340
 compactness_mean  concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
 Min.   :0.01938   Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
 1st Qu.:0.06492   1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
 Median :0.09263   Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154
 Mean   :0.10434   Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
 3rd Qu.:0.13040   3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
 Max.   :0.34540   Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
   radius_se        texture_se      perimeter_se       area_se        smoothness_se      compactness_se
 Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252
 1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080
 Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450
 Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478
 3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450
 Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400
  concavity_se     concave.points_se   symmetry_se       fractal_dimension_se  radius_worst   texture_worst
 Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02
 1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08
 Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41
 Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68
 3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72
 Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
 perimeter_worst    area_worst     smoothness_worst  compactness_worst concavity_worst  concave.points_worst
 Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000
 1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493
 Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993
 Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461
 3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140
 Max.   :251.20   Max.   :4254.0   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
```

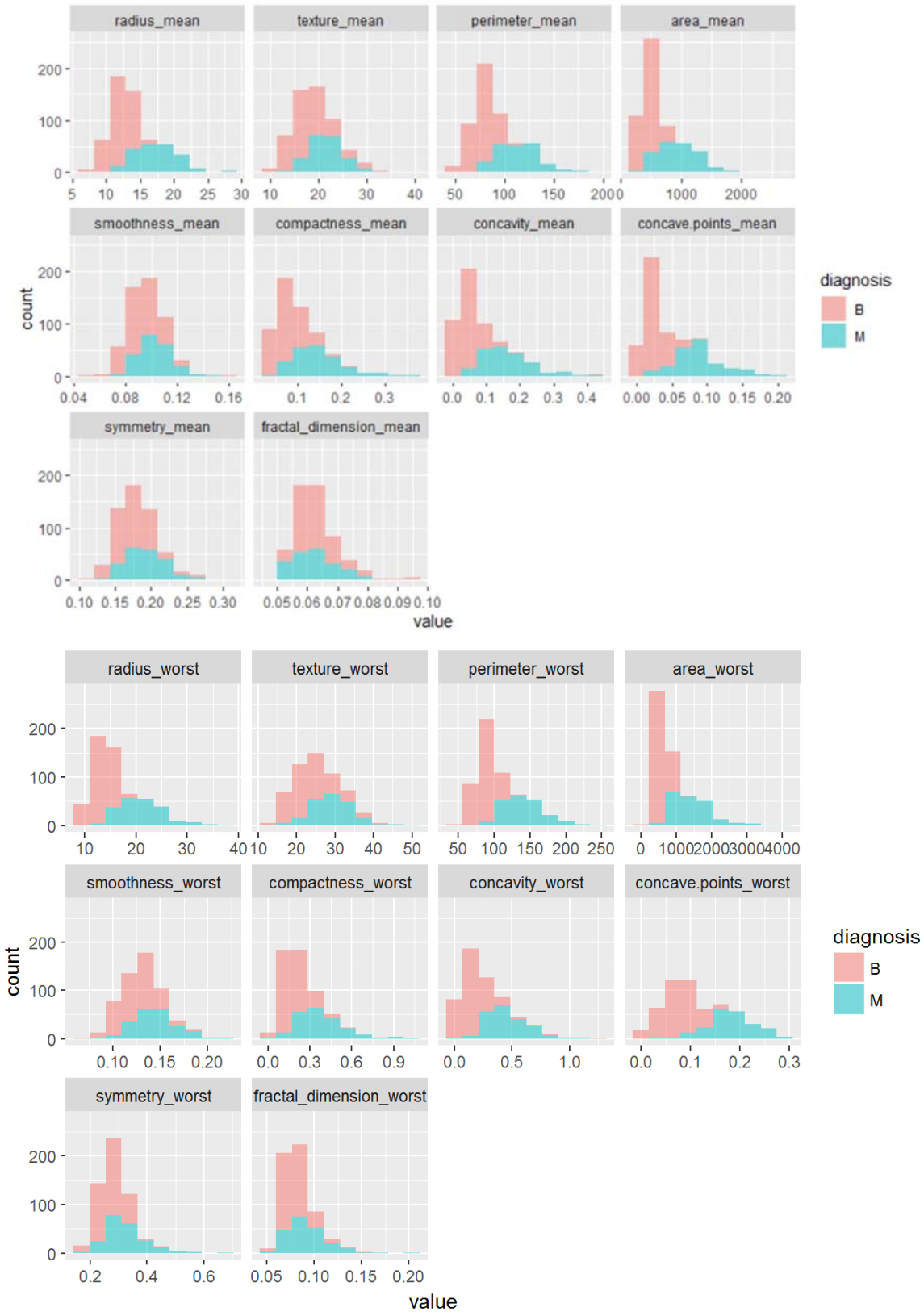


I first wanted to see the amount of data that was in the diagnosis column because that is the response variable that we are interested in. There are 357 cases of those who are benign (62.7%) and 212 of those who are malignant (37.3%). This is part of the data understanding phase of the data mining process. Furthermore, a histogram of the other variables were also done grouped by the type of variation it



was of the data (mean vs se vs worst).




radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave.points_mean
symmetry_mean
fractal_dimension_mean
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave.points_worst
symmetry_worst
fractal_dimension_worst
count
value
diagnosis
B
M

Data Preparation

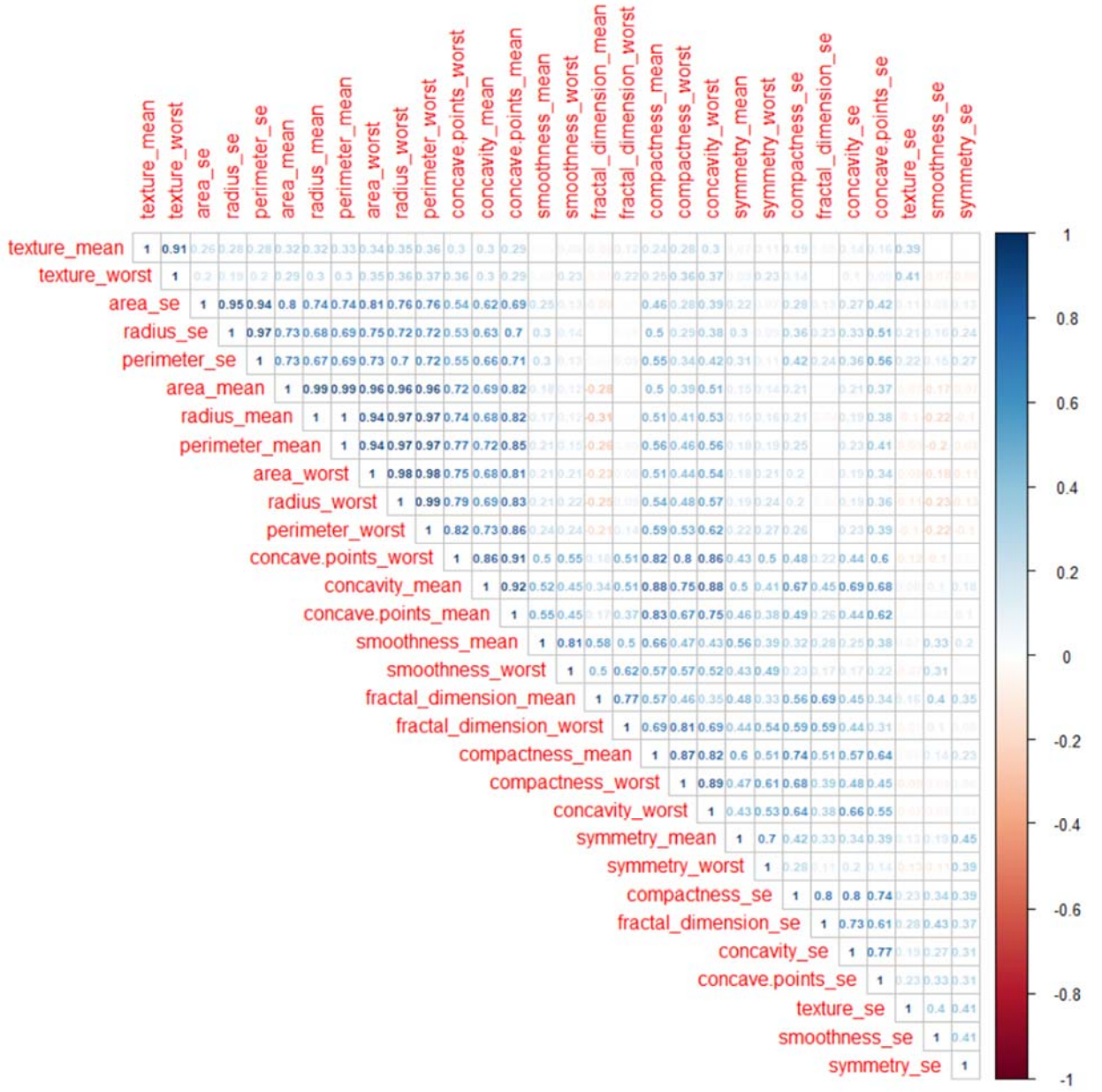
I needed to make sure that there were no missing values in the data. This was done by checking for missing values in R. Luckily, there were no missing values present within my data set.

```
> sapply(Data, function(x) sum(is.na(x)))
            diagnosis            radius_mean           texture_mean         perimeter_mean
                    0                      0                      0                      0
            area_mean         smoothness_mean        compactness_mean          concavity_mean
                    0                      0                      0                      0
   concave.points_mean           symmetry_mean  fractal_dimension_mean              radius_se
                    0                      0                      0                      0
           texture_se            perimeter_se                 area_se           smoothness_se
                    0                      0                      0                      0
        compactness_se            concavity_se       concave.points_se             symmetry_se
                    0                      0                      0                      0
  fractal_dimension_se            radius_worst           texture_worst         perimeter_worst
                    0                      0                      0                      0
            area_worst         smoothness_worst       compactness_worst         concavity_worst
                    0                      0                      0                      0
  concave.points_worst           symmetry_worst fractal_dimension_worst
                    0                      0                      0
```

Further checking of the variables were done by checking the correlation among all the variables with one another as shown on the correlation triangle shown on the next page. Many variables were seen to be correlated. This is not good because they would be providing redundant information. All those pairs with correlation higher than 0.9 were compared and the ones with a lower mean were eliminated from the data. The ones that were removed are shown below under the group “highlyCor”.

```
> highlyCor
 [1] "compactness_mean"      "concavity_mean"        "texture_worst"         "fractal_dimension_se"
 [5] "texture_mean"          "perimeter_worst"       "diagnosis"             "texture_se"
 [9] "perimeter_se"          "radius_mean"
```

We are down to 21 columns in our data set from the initial 32 columns of data. There was no change to the number of observations (rows).



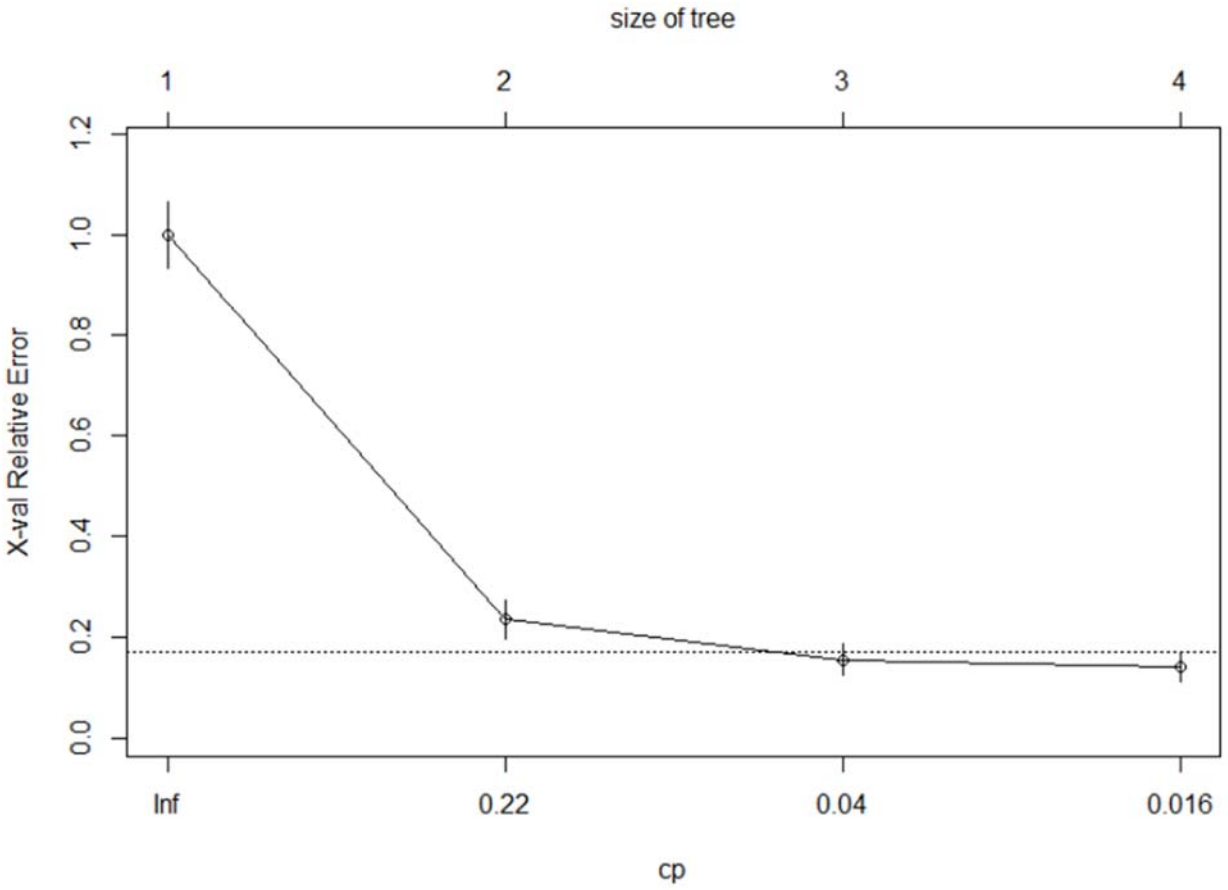
I split the data into p = 0.7 for the training set (399 observations) and the remaining data points were put in the testing set (170 observations). I wanted to try out all the classification models we learned in lab so I tried making a decision tree and a 5-fold cross validation method to train my model in the k-nearest neighbors model.

Modeling

I wanted to try the decision tree and k-nearest neighbors models because those are the ones we learned in lab for classification. For the decision tree model, the following variables were deemed the most important. The ones incorporated into my tree were concave.points_mean, area_worst, and radius_worst. Both the decision tree and k-nearest neighbors (seen few pages below) models both thought the same five variables were the most important even though they

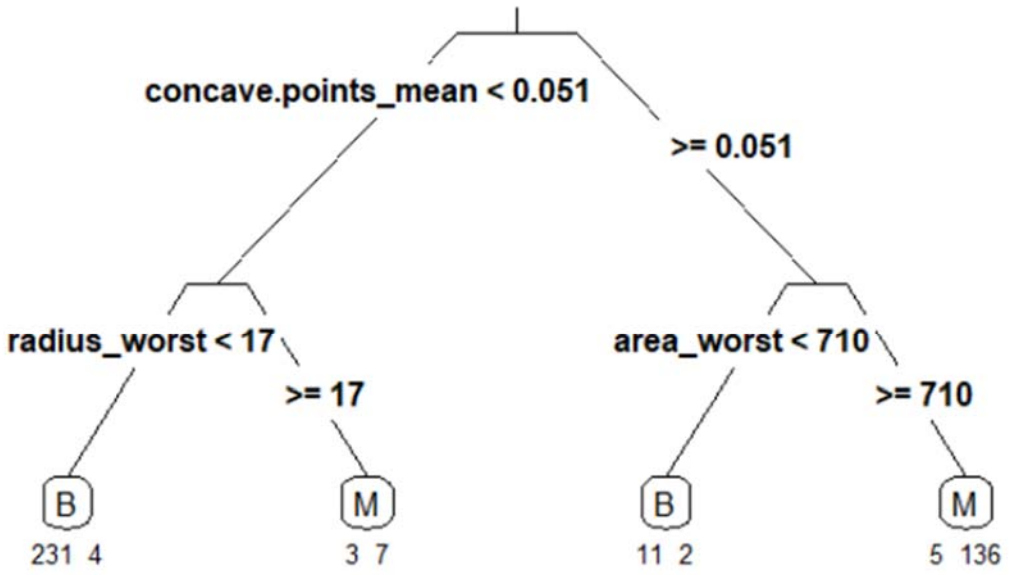
were not in the same order of importance. But due to the similar results that was gotten from both models, it makes sense they both thought certain variables were the most important.

```
> tree$variable.importance
   concave.points_mean             area_worst          radius_worst   concave.points_worst
             137.0292224            119.8023499            116.1918916            114.7842188
        perimeter_mean              area_mean                area_se          concavity_se
             106.9087811            105.1392635              6.6036274              3.6104583
fractal_dimension_mean
               0.8948415
```



```
> printcp(tree)

Classification tree:
rpart(formula = diagnosis ~ ., data = train_set)

Variables actually used in tree construction:
[1] area_worst          concave.points_mean radius_worst

Root node error: 149/399 = 0.37343

n= 399

        CP nsplit rel error  xerror     xstd
1 0.818792      0   1.00000 1.00000 0.064847
2 0.060403      1   0.18121 0.23490 0.037924
3 0.026846      2   0.12081 0.15436 0.031245
```

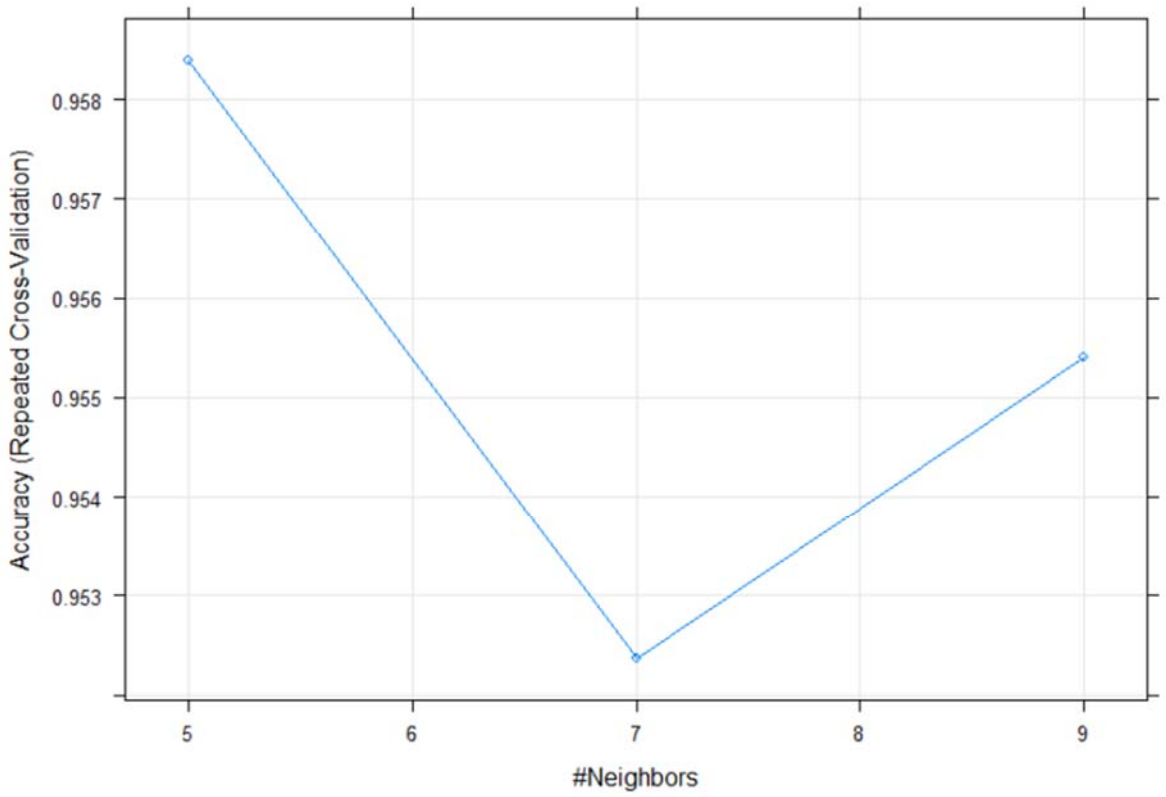
The size of the tree was decided at 3 due to the above picture. The picture to the right helped set the CP level to 0.025 so that up to nsplit 2 level could be included for pruning the tree.



This was the actual tree model that was made by the program. It used the three variables of most importance as stated above. In the

In the k-nearest neighbors model, I found out that when k = 5, it provided the most accuracy for my model.

```
k-Nearest Neighbors

399 samples
 21 predictor
  2 classes: 'B', 'M'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 5 times)
Summary of sample sizes: 319, 319, 320, 319, 319, 319, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.9583861  0.9090765
  7  0.9523734  0.8961427
  9  0.9553987  0.9026763

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```



```
> varImp(knn1)
ROC curve variable importance

  only 20 most important variables shown (out of 21)

                        Importance
concave.points_worst       100.000
radius_worst                99.892
area_worst                  99.587
concave.points_mean         98.325
perimeter_mean              94.737
area_mean                   92.466
concavity_worst             91.972
area_se                     89.620
compactness_worst           80.769
radius_se                   77.443
concave.points_se           62.719
concavity_se                62.271
smoothness_worst            53.650
symmetry_worst              53.459
compactness_se              51.054
smoothness_mean             44.797
symmetry_mean               44.428
fractal_dimension_worst     43.462
symmetry_se                 10.272
smoothness_se                2.878
```

The most important variables in this model are seen to the left.

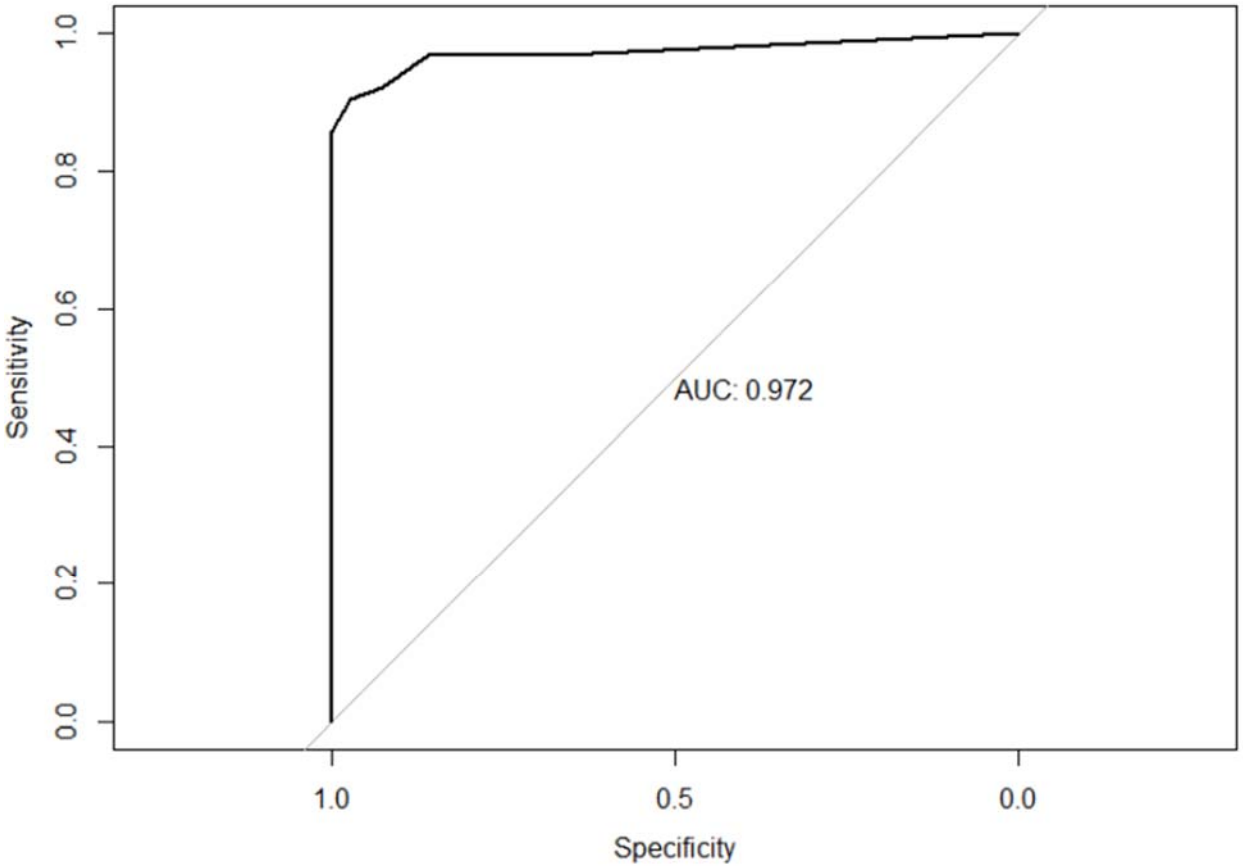
Evaluation

The two models performed admirably but the k-nearest neighbors ended up being the better model than the decision tree. Hyper-parameter tuning did not help in the case of the decision tree because I came up with the same results with and without pruning. I wanted to measure specificity because we do not want to miss any of those who might have the disease even if we get some false positives. The k-nearest neighbors was not only better in specificity, but also in sensitivity and overall accuracy. I think for this dataset, k-nearest neighbors modeling technique is favored. Both models had some malignant tumors predicted as benign. Surprisingly, the k-nearest neighbors did not mark any tumors as malignant when they were benign. This was not the case for the decision tree model.

```
> confusionMatrix(tree.pred, test_set$diagnosis)
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 100  10
         M   7  53

               Accuracy : 0.9
                 95% CI : (0.8447, 0.9407)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : 1.01e-15

                  Kappa : 0.7835

 Mcnemar's Test P-Value : 0.6276

            Sensitivity : 0.9346
            Specificity : 0.8413
         Pos Pred Value : 0.9091
         Neg Pred Value : 0.8833
             Prevalence : 0.6294
         Detection Rate : 0.5882
   Detection Prevalence : 0.6471
      Balanced Accuracy : 0.8879

       'Positive' Class : B
```

```
> confusionMatrix(pred, testing$diagnosis)
Confusion Matrix and Statistics

          Reference
Prediction   B   M
         B 107   7
         M   0  56

               Accuracy : 0.9588
                 95% CI : (0.917, 0.9833)
    No Information Rate : 0.6294
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9097

 Mcnemar's Test P-Value : 0.02334

            Sensitivity : 1.0000
            Specificity : 0.8889
         Pos Pred Value : 0.9386
         Neg Pred Value : 1.0000
             Prevalence : 0.6294
         Detection Rate : 0.6294
   Detection Prevalence : 0.6706
      Balanced Accuracy : 0.9444

       'Positive' Class : B
```

The results of the decision tree are on the left and the k-nearest neighbors results are on the right. You can see the better results for the k-nearest neighbors in almost every category measured.



The k-nearest neighbors model had a great AUC of 0.972. Being so close to one, this shows that the model showed great separability in being able to distinguish between the benign and malignant tumors.

Discussion and Conclusions

Both models were able to come up with a middle 80 to high 80 percent specificity for breast cancer detection. I would hope that the specificity number is much higher in the real world so that you do not miss many malignant tumors which could result in the loss of lives of loved ones. Getting some more data points and seeing if the model could become more accurate for decision trees and k-nearest neighbors would be something I would like to do. My guess is that the more observations there are, it could only help, if they truly are real observations. Working with real data on material that we went over in class, such as the decision tree and k-nearest neighbors models, was not daunting at all and we have learned a lot in the past 10 weeks of school!

References

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/activity
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
https://rstudio-pubs-static.s3.amazonaws.com/344010_1f4d6691092d4544bfbddb092e7223d2.html
https://shirinsplayground.netlify.com/2018/06/intro_to_ml_workshop_heidelberg/