Breast Cancer Research

John Kraft

5/5/2020

1. Project Overview and Objectives

The main objective of this project is to build multiple models that use the data's numerical values, that describe the geometric shape of a breast tumor, to predict if the subject's tumor is malignant. There are two classifications of a breast tumor that are malignant, which I mentioned before which means cancerous, and benign, which means the tumor is non-cancerous. It's important to accurately classify whether a breast tumor is cancerous or not because Malignant tumors can become extremely dangerous if it's gone untreated. Malignant tumors have the potential to invade their surrounding tissue or spread around one's body. Once the cancerous tissues invade neighboring tissues and spreads, it becomes difficult to contain/control the spread resulting in devastation. The three main metrics used to classify performance for models are accuracy, recall, and precision. The main metrics I will use to justify my models are accuracy and recall. Accuracy is defined as the number of correctly predicted subjects over the total number of subjects. A recall score is defined as the total number of true positives divided by the addition of true positive and false negatives. It can also be described as the number of how many people are told they don't have breast cancer when in reality they do. I put a higher importance on my recall compared to accuracy and precision because it can become deadly if a person is told they don't have breast cancer and they move on without seeking necessary treatment. I decided to split the data into training and test data sets. The training data set is used to train the models and the test data set is used to validate the performance of the models. The data was built by the International Symposium on Electronic Imaging: Science and Technology and consistence of 569 observations, 212 malignant, and 357 benign("Wisconsin Diagnostic Breast Cancer," n.d.). Also, the data does not consist of any missing values which help with setting up the environment.

1.1 Data Set Description

The Categorical variable I am trying to predict is currently encoded as M for malignant and B for benign, but I decided to encode the malignant tumors as 1's and the benign tumors as 0's. There are a total of 27 features that describe the geometric shape of the breast tumor. Shown below displays the data dictionary that explains the meanings of each feature.

1. radius_mean: mean of distances from center to points on the perimeter
2. texture_mean: standard deviation of gray-scale values
3. perimeter_mean: mean size of the core tumor
4. area_mean: mean area of the core tumor
5. smoothness_mean: mean of local variation in radius lengths
6. compactness_mean: mean of perimeter^2 / area - 1.0
7. concavity_mean: mean of severity of concave portions of the contour
8. concave points_mean: mean for number of concave portions of the contour
9. symmetry_mean: mean of spatial relationship amount the tumor
10. fractal_dimension_mean: mean for "coastline approximation" - 1
11. radius_se: standard error for the mean of distances from center to points on the perimeter

12. texture_se: standard error for standard deviation of gray-scale values
13. perimeter_se: standard error for the size of the core tumor
14. area_se: standard error for the area of the tumor
15. smoothness_se: standard error for local variation in radius lengths
16. compactness_se: standard error for perimeter^2 / area - 1.0
17. concavity_se: standard error for severity of concave portions of the contour
18. concave points_se: standard error for number of concave portions of the contour
19. symmetry_se: standard error for the spatial relationship of the tumor
20. fractal_dimension_se: standard error for "coastline approximation" - 1
21. radius_worst: "worst" or largest mean value for mean of distances from center to points on the perimeter
22. texture_worst: "worst" or largest mean value for standard deviation of gray-scale values
23. perimeter_worst: "worst" or largest mean value for the perimeter distance from the tumor's core
24. area_worst: "worst" or largest mean value for the tumor's area
25. smoothness_worst: "worst" or largest mean value for local variation in radius lengths
26. compactness_worst: "worst" or largest mean value for perimeter^2 / area - 1.0
27. concavity_worst: "worst" or largest mean value for severity of concave portions of the contour

1.2 What is a Cancerous Breast Tumor?

As a motivation to why this data is important to research and the applications machine learning models can be used for in the medical field, it empirical to understand what breast cancer is and how integrated it is in countless lives. A breast tumor occurs when abnormal cells form within the breast. A breast tumor can be classified as either malignant or benign which I describe the difference between the two above. Malignant tumors can be divided into primary tumors, which first developed in the breasts, and secondary tumor, which spread from elsewhere. About 1 in 8 women will develop breast cancer throughout her lifetime. It is extremely more common in women compared to men because a man's lifetime risk of breast cancer is roughly 1/883. It's critical to understand the urgency to research breast cancer because it is the second most common cancer (*U.S. Breast Cancer Statistics* 2020). Now there is a general understanding of what the data is consisted of and the concern for research, its time to move into setting up the environment to build models.

2 Setting up the Environment and Data Processing

```
df = read.csv("C:/Users/storm/Desktop/usingR (1)/usingR/CancerResearch_dataset.csv")
```

2.1 Cleaning the data

There was some feature that were not important and unnecessary to include in my analysis.

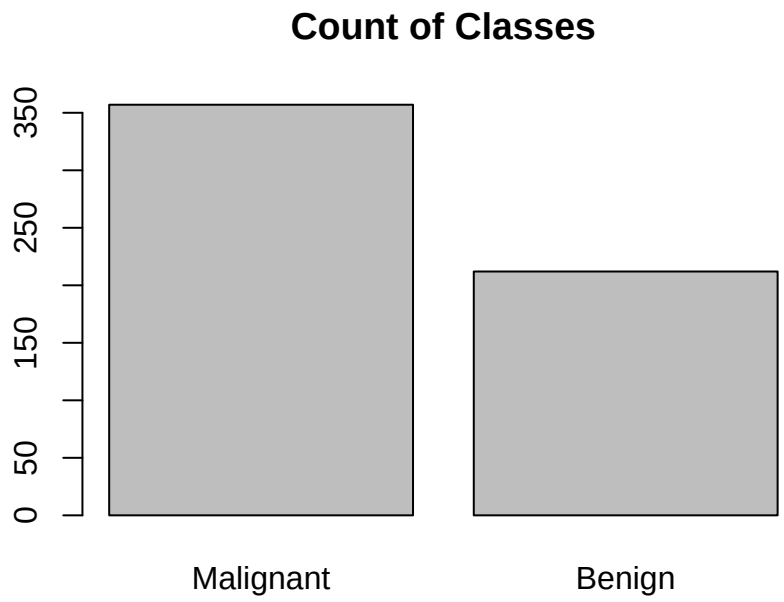
```
df <- df[, 2:29]
head(df,2)
```

```
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1         M       17.99        10.38          122.8      1001         0.11840
## 2         M       20.57        17.77          132.9      1326         0.08474
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001             0.14710        0.2419
## 2          0.07864         0.0869             0.07017        0.1812
##   fractal_dimension_mean radius_se texture_se perimeter_se area_se
```

```
## 1                0.07871    1.0950     0.9053        8.589  153.40
## 2                0.05667    0.5435     0.7339        3.398   74.08
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 1      0.006399        0.04904      0.05373           0.01587     0.03003
## 2      0.005225        0.01308      0.01860           0.01340     0.01389
##   fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## 1             0.006193        25.38         17.33           184.6       2019
## 2             0.003532        24.99         23.41           158.8       1956
##   smoothness_worst compactness_worst concavity_worst
## 1           0.1622            0.6656          0.7119
## 2           0.1238            0.1866          0.2416
```

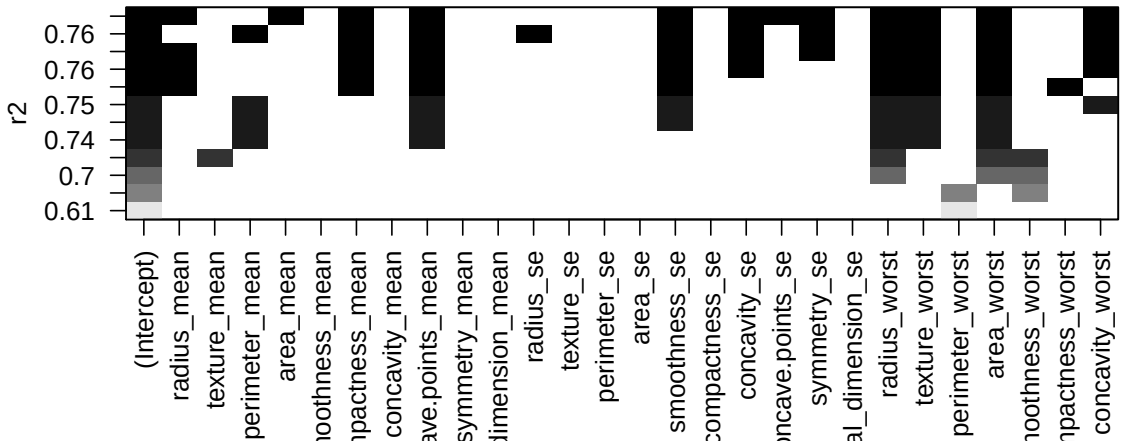
Displaying the difference in counts between malignant and benign tumors.

```
counts <- table(df$diagnosis)
barplot(counts, main="Count of Classes", names.arg=c("Malignant", "Benign"))
```



2.2 Looking into the data and subsetting the data

```
regfit <- regsubsets(diagnosis~.,df)
regfit <- regsubsets(diagnosis~.,data=df,nvmax=12)
plot(regfit, scale="r2")
```



Based of the subsetted data it is noticeable that the compactness_mean, concave.points_mean, smoothness_se, radius_worst, texture_worst, area_worst, and the concavity_worst are the best features to include. Therefore, I subsetting the data into a new dataframe that only included these features. Also, I must encode the diagnosis as 0s and 1s.

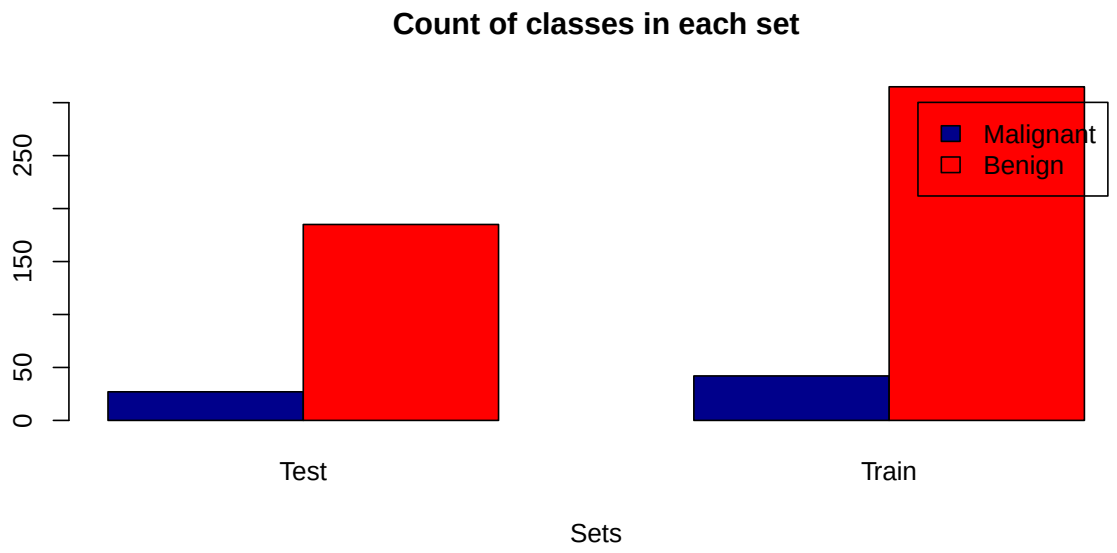
```
newdf <- select(df, "diagnosis", "compactness_mean", "concave.points_mean", "smoothness_se",
                "radius_worst", "texture_worst", "area_worst", "concavity_worst")
newdf$diagnosis <- ifelse(newdf$diagnosis=='M',1,0)
head(newdf, 2)
```

```
##   diagnosis compactness_mean concave.points_mean smoothness_se radius_worst
## 1         1          0.27760             0.14710      0.006399        25.38
## 2         1          0.07864             0.07017      0.005225        24.99
##   texture_worst area_worst concavity_worst
## 1         17.33       2019          0.7119
## 2         23.41       1956          0.2416
```

2.2 Splitting the data into training and testing data set

```
train <- sample(1:nrow(newdf), 500)
traindata <- newdf[train,]
testdata <- newdf[-train,]

tab <- matrix(c(27, 42, 185, 315), ncol=2, byrow=T)
colnames(tab) <- c("Test", "Train")
rownames(tab) <- c("Malignant", "Benign")

tab <- as.table(tab)

barplot(tab, main="Count of classes in each set",
```

```
  xlab="Sets", col=c("darkblue","red"),
  legend = T, beside=T)
```



Now that I have my training and test data all set up, its time to develop a hypothesis and decide which models I plan on using to conclude my hypothesis. Also, determine what am I exactly trying to predict and how to dictate whether my models are up to the satisfaction standard.

3. Developing a Hypothesis

I will state a few hypotheses to determine whether there is a relationship between two variables.

1. There is no relationship between the diagnosis and the compactness_mean.
2. There is no relationship between the diagnosis and the concave.points_mean.
3. There is no relationship between the diagnosis and the smoothness_se.
4. There is no relationship between the diagnosis and the radius_worst.
5. There is no relationship between the diagnosis and the texture_worst.
6. There is no relationship between the diagnosis and the area_worst.
7. There is no relationship between the diagnosis and the concavity_worst.

Next I will try to reject these null hypotheses by running multiply t-tests only comparing individual variables

```
t.test(newdf$diagnosis, newdf$compactness_mean)
```

```
##
##  Welch Two Sample t-test
##
## data:  newdf$diagnosis and newdf$compactness_mean
## t = 13.144, df = 581.53, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  0.2281614 0.3083236
## sample estimates:
## mean of x mean of y
## 0.3725835 0.1043410
```

```
#t.test(newdf$diagnosis, newdf$concave.points_mean)
#t.test(newdf$diagnosis, newdf$smoothness_se)
#t.test(newdf$diagnosis, newdf$radius_worst)
#t.test(newdf$diagnosis, newdf$texture_worst)
#.test(newdf$diagnosis, newdf$area_worst)
#t.test(newdf$diagnosis, newdf$concavity_worst)
```

After running all t.tests I was able to reject all my null hypotheses and therefore conclude there is a relationship between the diagnosis and the above variables. In the below table it displays the p-value for each feature amount the subsets data. Also, each feature can be classified as highly significant. Therefore, all these variables are suitable for building prediction models to dictate whether a breast tumor is cancerous or not.

```
DT = data.table(
  Results = "P-Value",
  "compactness_mean" = '2.2e-16',
  "concave.points_mean" = '2.2e-16',
  "smoothness_se" = '2.2e-16',
  "radius_worst" = '2.2e-16',
  "texture_worst" = '2.2e-16',
  "area_worst" = '2.2e-16',
  "concavity_worst" = '6.391e-06'
)
DT
```

```
##    Results compactness_mean concave.points_mean smoothness_se radius_worst
## 1: P-Value          2.2e-16             2.2e-16       2.2e-16      2.2e-16
##    texture_worst area_worst concavity_worst
## 1:       2.2e-16    2.2e-16       6.391e-06
```

4 Developing and testing Models

The next step of the process after ensuring I am using the correct features in the data that mostly relate to the diagnosis of the breast tumor is to build and test various models to correctly predict whether a tumor is cancerous or not. Since all the variables are numerical, I specifically chose models that would perform the best with numerical variables that predict a categorical variable. I chose three models to analyze, K-Nearest Neighbors, decision tree, and random forest. For KNN and the decision tree, I will calculate the accuracy and the recall score, as well as build a confusion matrix between the predicted diagnosis of the test data compared to the actual diagnosis of the test data. For the random forest, I will calculate the mean error and use it as my model's performance metric.

First, I want to create functions that receives tables and returns the accuracy and recall score.

```
acc <- function(tab){
  return((tab[1]+tab[4])/sum(tab))
}
```

```
recall <- function(tab){
  return(tab[4]/(tab[2]+tab[4]))
}
```

4.1 K-Nearest Neighbor

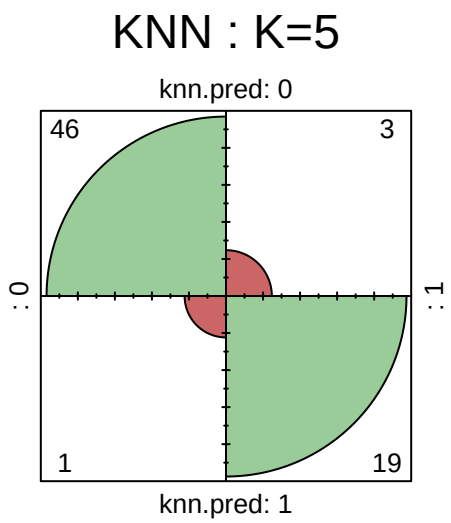
For my first model, I will be using one of the simplest machine learning algorithms out there which is K-Nearest Neighbor. Essentially, it stores all available cases and predicts new cases based on a similarity measure. The variable I will be changing is the number of neighbors each case has to dictation what is classified as, malignant or benign. Through all the KNN models I chose to only use odd numbers neighbors so there isn't a tie and need for a random decision.

First, I will build a KNN model where K=3 and train the model with my training data set. Then I will test my accuracy and recall score by building a confusion matrix.

```
set.seed(1)

knn.pred=knn(traindata,testdata,traindata$diagnosis,k=3)
keq3<-table(knn.pred, testdata$diagnosis)

knn.pred=knn(traindata,testdata,traindata$diagnosis,k=5)
keq5<-table(knn.pred, testdata$diagnosis)

knn.pred=knn(traindata,testdata,traindata$diagnosis,k=7)
keq7<-table(knn.pred, testdata$diagnosis)

knn.pred=knn(traindata,testdata,traindata$diagnosis,k=9)
keq9<-table(knn.pred, testdata$diagnosis)

scoretable <- matrix(c(acc(keq3),acc(keq5),acc(keq7),acc(keq9),
                       recall(keq3),recall(keq5),recall(keq7),recall(keq9)),
                     ncol=4,byrow = T)
colnames(scoretable) <- c("K=3","K=5","K=7", "K=9")
rownames(scoretable) <- c("Accuracy","Recall")
scoretable <- as.table(scoretable)
scoretable
```

```
##                K=3       K=5       K=7       K=9
## Accuracy 0.9275362 0.9420290 0.9420290 0.9420290
## Recall   0.9473684 0.9500000 0.9500000 1.0000000
```

```
fourfoldplot(keq5, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "KNN : K=5")
```

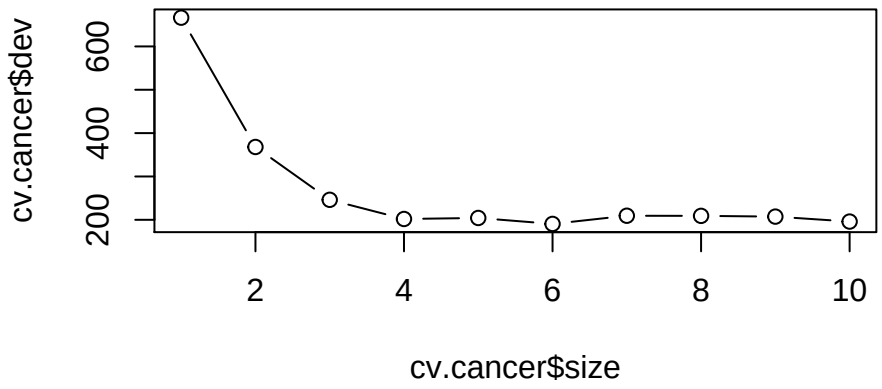


After trying k=3, k=5, k=7, and k=9, I discover the accuracy improving after k=5 and would range from 88-94% which is very good for a simple model. Next, I analyzed the recall score which would usually halt at k=5 ranging from 88-96. I'm okay with the recall score typically being higher than the accuracy score because that just means there is a lower precision score than recall. It is better to diagnosis someone with cancer who doesn't actually have it compared to tell someone they don't have cancer when in reality they do. One thing to not is I noticed the KNN model is very inconsistent and hard to rely on because for some model the accuracy was north of 95% and the recall landed at 100% but for other times the accuracy struggled to get about 92% and the recall had a tough time getting about the 90 percentile. In the figures above, they display the difference in accuracy and recall score compared between the k values, along with a confusion matrix of the ideal KNN model where K=5. Next, I will try to build a better model with decision trees compared to KNN.

4.2 Decision Tree

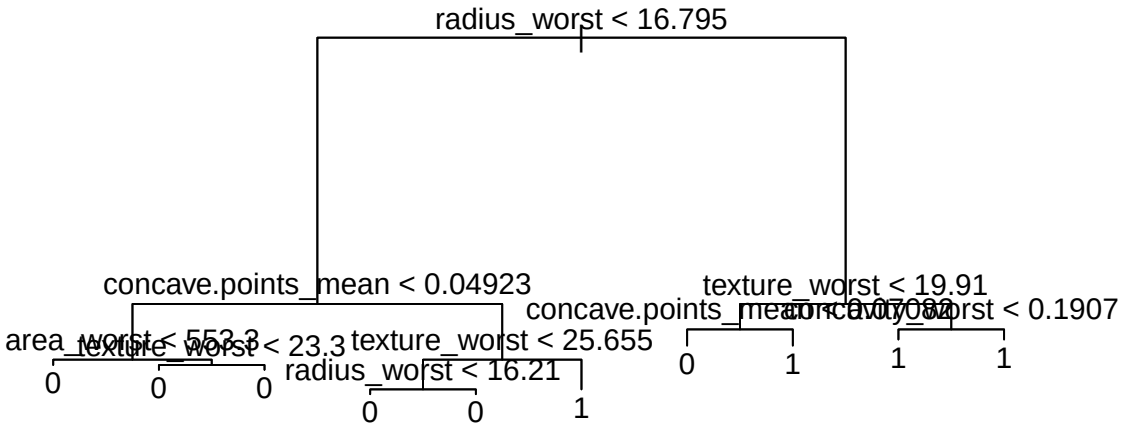
For my second type of model, I will be trying to use a decision tree to predict whether a tumor is malignant or benign. A decision tree has a flowchart-like structure where each node splits off into two paths and is decided by a specific feature. Also, each leaf of the tree represents the diagnosis. I will train my tree using my train data and calculate the accuracy and recall score by predict on the test data.

```
traindata$diagnosis <- as.factor(traindata$diagnosis)
tree.cancer <- tree(diagnosis~.-diagnosis,traindata)

cv.cancer <- cv.tree(tree.cancer)
plot(cv.cancer$size,cv.cancer$dev,type='b')
```

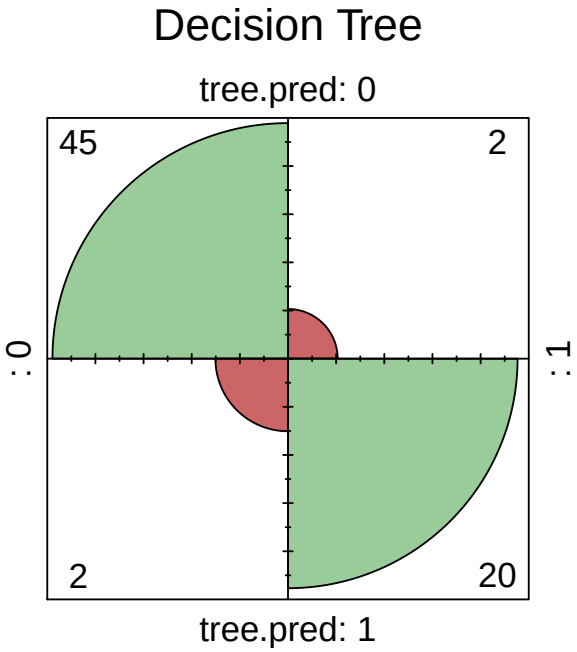


After plotting the cross-validation score for each score, it is apparent that the best number of features should be five. Therefore, I decided it would be best to prune the tree. Even though for some decision trees, it is ideal to include all the features, it doesn't improve much more from five features to ten features if at all. So, its not worth the extra resources to develop the model with 10 features over five. Also, by reducing the number of features used, it reduces the risk of overfitting the training data.

```
prune.cancer <- prune.tree(tree.cancer, best=5)
plot(tree.cancer)
text(tree.cancer,pretty=2)
```



```
tree.pred <- predict(prune.cancer,testdata, type="class")
decTree <- table(tree.pred,testdata$diagnosis)

fourfoldplot(decTree, color = c("#CC6666", "#99CC99"),
             conf.level = 0, margin = 1, main = "Decision Tree")
```




```
scoretable <- matrix(c(acc(decTree),recall(decTree)),ncol=1,byrow = T)
colnames(scoretable) <- c("Tree")
rownames(scoretable) <- c("Accuracy","Recall")
scoretable <- as.table(scoretable)
scoretable
```

```
##               Tree
## Accuracy 0.9420290
## Recall   0.9090909
```

From the results above about decision trees, I can conclude that a decision tree consistently has higher accuracy and recall scores than the KNN models I built. Even though I am using numeric values for my nodes, it still predicts very well and just uses whether a value is greater than or less than the value to decide which path to take. The downfall of a decision tree is they are known for overfitting data, so it is difficult to know how consistent the accuracy and recall scores are if I were to scale the data up much higher.

4.3 Random Forest

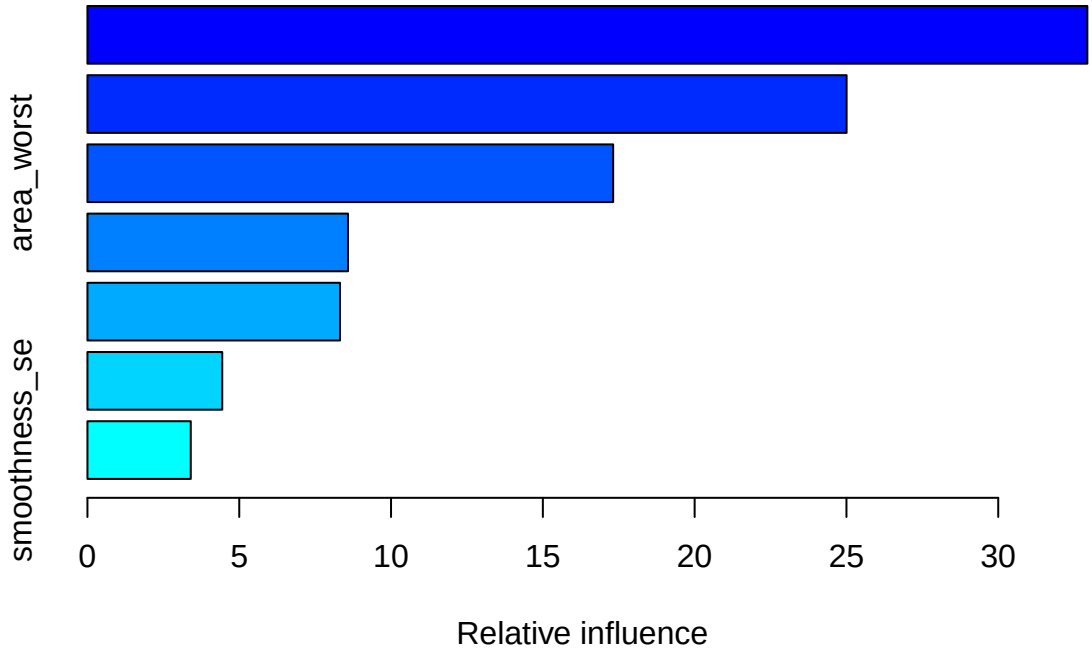
The final I chose to explore was random forests. A random forest is an ensemble learning and uses multiple decision trees who work together to build a random forest model. I thought it would be fascinating to build a random forest to see how well it can do to my other two models because it is easy to understand how

decision trees and KNN models are built, but a random forest is sort of a black box and difficult to visualize how it is built.

```
set.seed(1)
names(traindata)
```

```
## [1] "diagnosis"           "compactness_mean"    "concave.points_mean"
## [4] "smoothness_se"       "radius_worst"        "texture_worst"
## [7] "area_worst"          "concavity_worst"
```

```
boost.cancer=gbm(diagnosis~.,data=traindata,distribution="gaussian",n.trees=5000,interaction.depth=4)
summary(boost.cancer)
```




```
##                                   var   rel.inf
## concave.points_mean concave.points_mean 32.934961
## radius_worst               radius_worst 25.004076
## area_worst                   area_worst 17.316584
## texture_worst             texture_worst  8.581202
## concavity_worst         concavity_worst  8.319443
## compactness_mean       compactness_mean  4.439966
## smoothness_se             smoothness_se  3.403767
```

Observe how the area_worse and the concave.point_mean are the most influential features for a random forest. Trying to understand why can give us a little insight into how cancer is diagnosed. For instance, if

there is an extreme dip in the tumor is can help diagnosis a tumor cancer. Also, if there is a rogue cell where its area is absurdly large, then it could lead us in the correct direction of diagnosing a patient with cancer. Next, I will predict the test data using this model and calculate the mean error of the model.

```
yhat.boost=predict(boost.cancer,newdata=testdata,n.trees=5000)
x <- mean((yhat.boost-as.numeric(testdata$diagnosis))^2)
x-1
```

```
## [1] 0.0659333
```

The mean error of the model is extremely low which is great because that means the accuracy of the model is high. It is difficult to compare the accuracy and recall scores calculated in the KNN and decision tree models to the mean error in the random forest, but a mean error of less than 5% is highly reliable. Therefore, in conclusion, I would classify the random forest model as the ideal model out of these three to predict whether a tumor is malignant or benign based on the tumor's geometric features.

Conclusion

One of my high goals was to achieve high accuracy and recall scores or accomplish a small mean error value by building these various models and using the geometric shape of a breast tumor to predict whether it was malignant or benign. Overall, I was happy with my results of developing a KNN, decision tree, and random forest model where all of them seemed to stay above the 90% accuracy and 90% recall score. It is still important to consider these models would never be allowed to be used in the medical field because 1/10 misdiagnosis is way too high, especially a 1/10 misdiagnosis resulting in a false negative that could end drastically poor. For now, these models should only be used to try to get a deeper understanding of how machine learning can have applications in the medical field especially diagnosis. Building each model was fascinating because they are all fairly different models, yet they all set on the same key features. For instance, the radius worse and the concave.point_mean seemed to play a significant role in all the models. If I had the opportunity to continue my research I would research into seeing if I would be able to predict the mean_area of the tumor based on the other geometric features. This could have some application because even though benign tumors are not dangerous, if they are too big, they can pose a threat and the removal is unavoidable. In some other research, I analyzed MRI images with a convolutional neural network to predict whether the MRI image had a brain tumor in it or not. This led me to start thinking if I could use the model I build with breast tumors, to predict whether the brain tumor is cancerous or not. Do all malignant and benign tumors carry similar traits? Another model I would use, that I wasn't able to get to in this analysis, is support vector machines. SVMs are known to be very powerful when it comes to numerical data, so it would be fascinating if I would be able to do better with SVM compared to he KNN, decision tree, and random forest models I already built. There are numerous treatments for breast cancer, but no one has discovered a 100% survival rate treatment for all breast tumors and breast cancer. This is the sad reality of millions of people around the world who live with every day. Even though there is no cure, there is an opportunity to correctly diagnose patients with breast cancer in the early stages with machine learning. Machine learning can create hope for the medical industry because if there is an algorithm that can correctly diagnosis patients at an extremely high rate or perfect rate, there may be an opportunity to save lives. This would allow doctors to focus on treatment and less on diagnosing because for some patients, time may be limited and every minute counts.

U.S. Breast Cancer Statistics. 2020. Pennsylvania, U.S.: Breastcancer.org. https://www.breastcancer.org/symptoms/understand_bc/statistics.

"Wisconsin Diagnostic Breast Cancer." n.d. *International Symposium on Electronic Imaging: Science and Technology* 1905: 861–70. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.