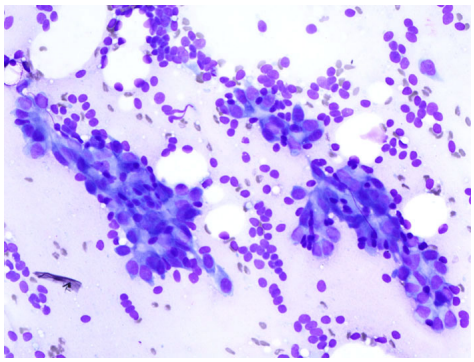
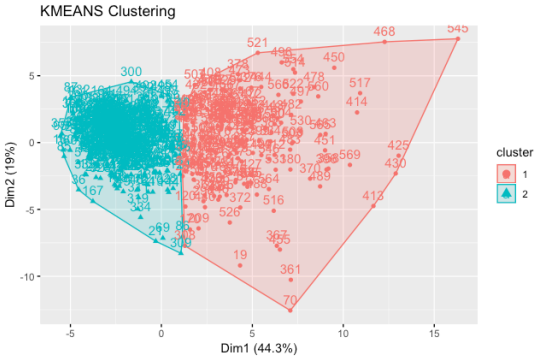
Using Radial SVM Models to Predict Breast Cancer Diagnoses From Fine Needle Aspiration Biopsy Data

Carlos Flores, May 3, 2019
Computer Science Department, University of Houston, Houston, TX 77004
cflores28@uh.edu





Source: Wikipedia Commons (Permission is granted under the terms of the GNU Free Documentation License).

Abstract

Recent studies have found that women with a history of a false-positive mammogram results may be at increased risk of developing subsequent breast cancer, but the origin of these regularities has remained opaque.[1] I analyzed the physical characteristics of biopsy data to develop various successful predictive models needed to for such regularities to emerge. The result is a Radial SVM model to help radiologists and pathologists to investigate abnormal cellular growth irrespective of preconceived demographic biases. The model efficiently leverages statistical information by training only on the feature space of the physical bio-markers. Therefore, creating a meaningful substructure with significantly less computational demand given the reduced dimensionality. Two other models, KMeans and KNN, performed similarly and give us greater insight into the emergent regularities.

[1] Henderson LM, Hubbard RA, Sprague BL, Zhu W, and Kerlikowske K: "Increased Risk of Developing Breast Cancer after a False-Positive Screening Mammogram", Cancer Epidemiol Biomarkers Prev, December 1 2015 (24) (12) 1882-1889; DOI: 10.1158/1055-9965.EPI-15-0623 link

1. Introduction:

The United States has a relatively high rate of false-positives for breast cancer screenings amongst industrialized nations. For instance, the Netherlands has a misdiagnosis rate of 1% while the US leads with 15%. After ten yearly mammograms, the chance of having a false positive is about 50-60 percent.[2] In order to investigate an abnormal finding on a mammogram the Netherland protocol requires a second mammogram and a biopsy, while the American protocol calls for a single biopsy. The stringent screening policies by the Netherlands is attributed as the reason for it's high success rate.

Given the inevitability of a biopsy to be performed it would be useful to look into the microscopy data as well. Previous clinical studies have looked only into the demographic, environmental, and historical factors to identify patterns. They typically converted quantitative features into categorical variables through arbitrary binning. I developed an evaluation scheme that incorporates the physical features of the cancer cells and kept the quantitative values as continuous features rather than discrete qualities.

The University of Wisconsin Clinical Sciences Center has made a data set publicly available in order to help predict whether a patient will develop breast cancer. The goal is to identify specific physical features from a Fine Needle Aspiration (FNA) biopsy to predict if a tumor is malignant or benign.[3]

The data set includes ten features that pertain to physical characteristics of the tumor cells. All of the predictors are measurements of the dimensions of the cells: such as the shape (concavity, concave points, fractal dimension, symmetry) or size (perimeter, radius, coastline approximation, area). The other two features measure the texture and smoothness.

The primary motivation is in seeing how the data can be used to predict the patient's diagnosis, and to identify key physiological markers for malignancy. This would give greater insight into the methods in which unregulated cellular growth can start and lead to cancer.

[2] Hubbard RA, Kerlikowske K, Flowers CI, Yankaskas BC, Zhu W, Miglioretti DL. Cumulative probability of false-positive recall or biopsy recommendation after 10 years of screening mammography: a cohort study. Ann Intern Med. 155(8):481-92, 2011.

[3]Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. link.

2. Methodology:

Given that the problem has a clearly defined response variable I sought to use Supervised Learning models (KNN and SVM) to evaluate my predictions. Additionally, I wanted to see how KMeans clustering can be interpreted with help from External Validation. I expect KMeans to cluster at two natural groupings. Since the data set has 563 observations in 10 dimensions either model is likely to perform well; it is not readily apparent yet. A few factors include:

Models Used

SVM is a Supervised Model, KNN is a Supervised Classification, and KMeans is an Unsupervised Clustering Classifier. When you introduce a new observation, the SVM model simply finds on what side of the hyperplane the observation exists. While, KNN and KMeans would require the entire function to recalculate the K nearest observations or group the observations into K discrete groupings respectively.

Dimensionality

SVM works great for a large amount of observations in a low dimensional space while KNN is the opposite. KMeans performs poorly as the dimensions grow. It is beneficial to use PCA to eliminate/truncate features that are relatively irrelevant.

Scaling

The KMeans models will use the data set with 10-predictors since clustering methods are known to perform better in feature spaces in lower dimensions. The data set will be scaled and compared with the raw data set.

Boundary Shape

SVM can only work to separate linearly separable data but can use kernel functions to extend the feature space to gain non-linearity. Fortunately, the features are already considered to be linearly separable as proven by Bennett and Mangasarian [4]

KNN and KMeans can create arbitrary non-linear boundaries easily.

[4] K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34

Error Prevention

My primary emphasis is to find a model with a relatively low amount of errors but some consideration will be devoted to preventing Type II/ False Negative errors. Failing to classify a patient's tumor as malignant (when it is indeed cancerous) would give the patient a false sense of security and risk their health. Especially since the cancer would have more time to develop into a more serious illness.

Priority Rationale

When presented with various models with similar rates of success the amount of Type II errors will be the second priority and the third priority will be complexity. We should aim to find a model that performs well and is efficiently successful. This would lead towards a model that is more feasible in an actual implementation professionally.

Explicit Optimization Task Formulae: K Nearest Neighbors:

$$Pr(Y = j \mid X = x_0) = \frac{1}{K} \sum_{i \in N_0} \mathbb{I}(Y_i = j)$$

Hyperplane Equation for a Radial Support Vector Machine:

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K\langle x_i, x_j \rangle \mid K\langle x_i, x_j \rangle = exp(-\gamma) \sum_{k=1}^{P} (X_{ik} - X_{jk})^2) \mid \gamma > 0$$

K Means Clustering Optimization Criteria:

$$\underset{c_1,\ldots,c_k}{minimize} \Big\{ \sum_{k=1}^{k} \frac{1}{|C_k|} \sum_{i,j \in C_k, i>j} \sum_{r=1}^{p} (x_{ir} - x_{jr})^2 \Big\}$$

Data Preprocessing and Considerations:

I arranged the dataset by the Diagnosis class label. The Benign patients were the first 371 observations and the Malignant patients were the last observations. The creators of the original dataset used 10 features and each of those features were expanded by two conditions; the Standard Error and the Worst Outlier. The Standard Error, $(x_{ib} - \mu_b)$, is the difference between

the i^{th} observation and the mean of that feature. While the Residual of the Worst Outlier, $(x_{ib} - x_{worst})$, is the difference between this i^{th} observation and the furthest outlier (labeled as x_{worst}).

Feature Space

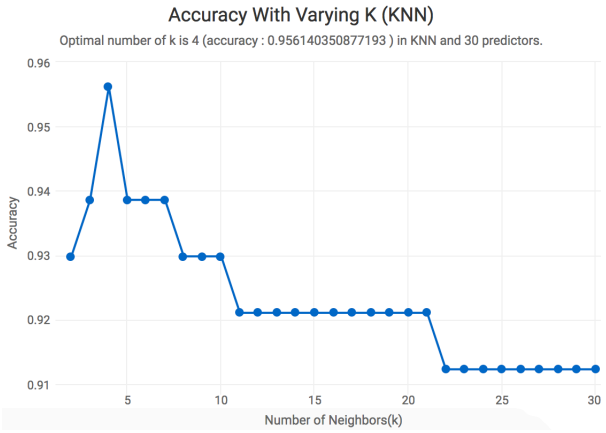
Therefore, the creators had 30 total predictors available (10 original predictors, 10 standard errors, 10 residuals from the worst observation). I threw the entire dataset into the three models: KNN, SVM, and KMeans. Additionally, I used a smaller dataset of only 10 features and disregarded the rest of the standard error/residual data. I was interested in seeing how the additional data points affected the performance of the models.

3. Model Performance and Tuning: K Nearest Neighbors

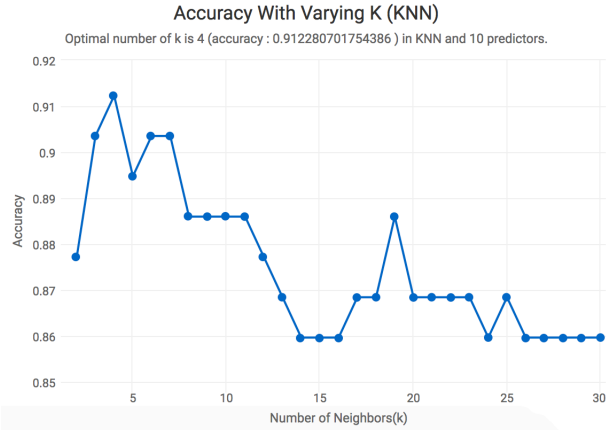
```
### KNN
#library(class)
#knn.acc is an array to store values
knn.acc <- numeric()
for(i in 2:30){
  set.seed(1)
  knn.predict <- knn(train=train[,-1], test=test[,-1], cl=train[,1], k=i, prob=T)
  #Store the average number of correctly classified points
  knn.acc <- c(knn.acc,mean(knn.predict==test[,1]))
}

project.acc <- data.frame(k= seq(2,30), cnt = knn.acc)
opt_k <- subset(project.acc, cnt==max(cnt))[1,]
subtitle <- paste("Optimal k:", opt_k$k,
                  "(Test Error :", 1-opt_k$cnt,").")

#library(magrittr)
#library(highcharter)
hchart(project.acc, 'line', hcaes(k, cnt)) %>%
  hc_title(text = "80/20 Split With 30 Predictors (KNN)") %>%
  hc_subtitle(text = subtitle) %>%
  hc_add_theme(hc_theme_google()) %>%
  hc_xAxis(title = list(text = "Number of Neighbors(k)")) %>%
  hc_yAxis(title = list(text = "Accuracy"))
```



KNN with 30 Predictors



KNN with 10 Predictors

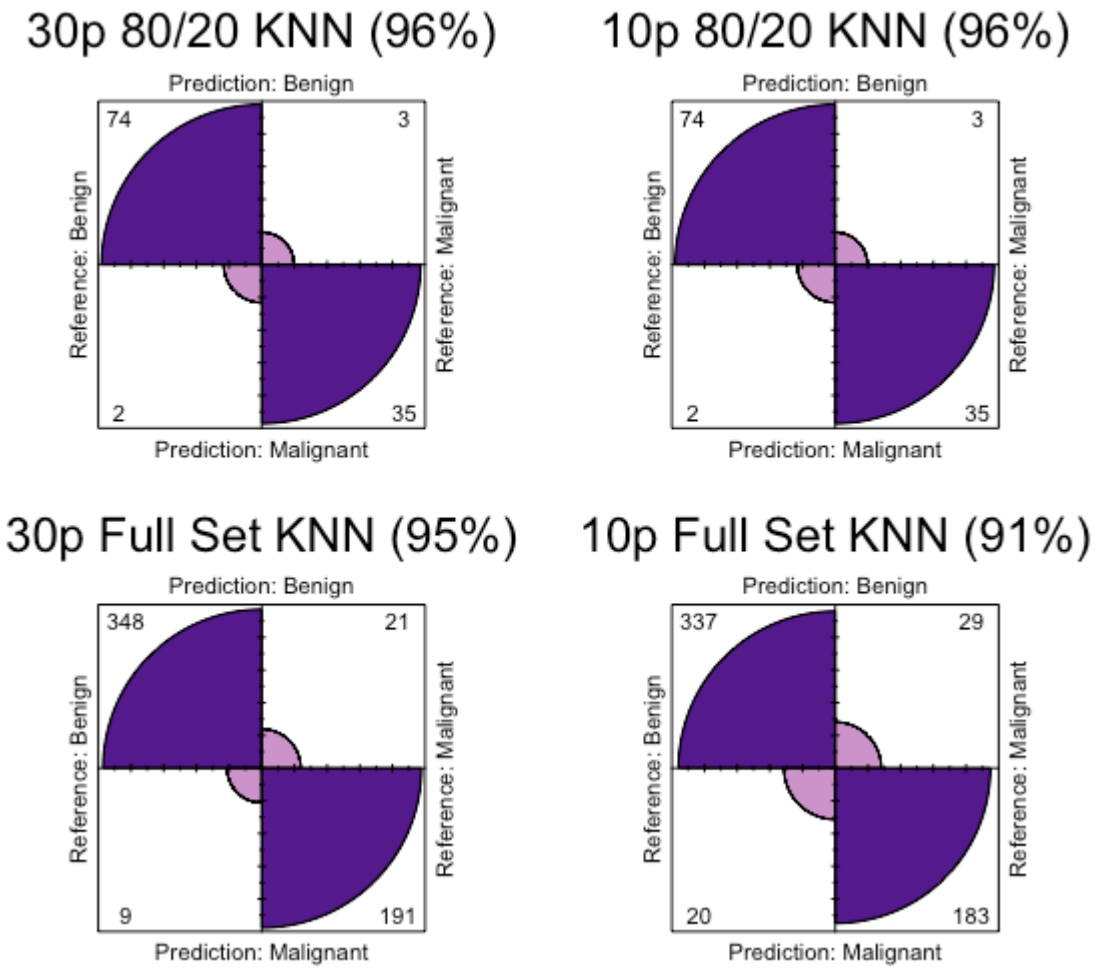
Both models worked best with K= 4 neighbors

```r
# 30 predictors 4 neighbors 80/20 split @ 96%
set.seed(1)
pre_knn <- knn(train = train[,-1], test = test[,-1],
               cl = train[,1], k=opt_k$k, prob=T)
cm_knn  <- confusionMatrix(pre_knn, test$diagnosis)
# 30 predictors with 4 neighbors Full Set @ 95%
set.seed(1)
knn.pre.30.full <- knn(train = wbcd[,-1], test = wbcd[,-1],
                       cl = wbcd[,1], k=opt_k$k, prob=T)
knn.cm.30.full  <- confusionMatrix(knn.pre.30.full, wbcd$diagnosis)
mean(knn.pre.30.full != wbcd[,1])
```

```r
# 10 predictors 10 neighbors @ 96%
set.seed(1)
knn.pre10 <-knn(train = train[,-1], test = test[,-1],
                cl = train[,1], k=opt_k.10$k, prob=T)
knn.cm.10  <- confusionMatrix(knn.pre10, test$diagnosis)
# 10 predictors with 4 neighbors Full Set @ 91%
set.seed(1)
knn.pre.10.full <-knn(train = wbcd10[,-1], test = wbcd10[,-1],
                      cl = wbcd10[,1], k=opt_k.10$k, prob=T)
knn.cm.10.full  <- confusionMatrix(knn.pre.10.full, wbcd10$diagnosis)
```

I obtained four KNN models by training with four neighbors (optimal K), by using using either a 80/20 Cross Validation split or the full set, also by using either 30 predictors or 10 predictors.

The results for the four models are as follows:

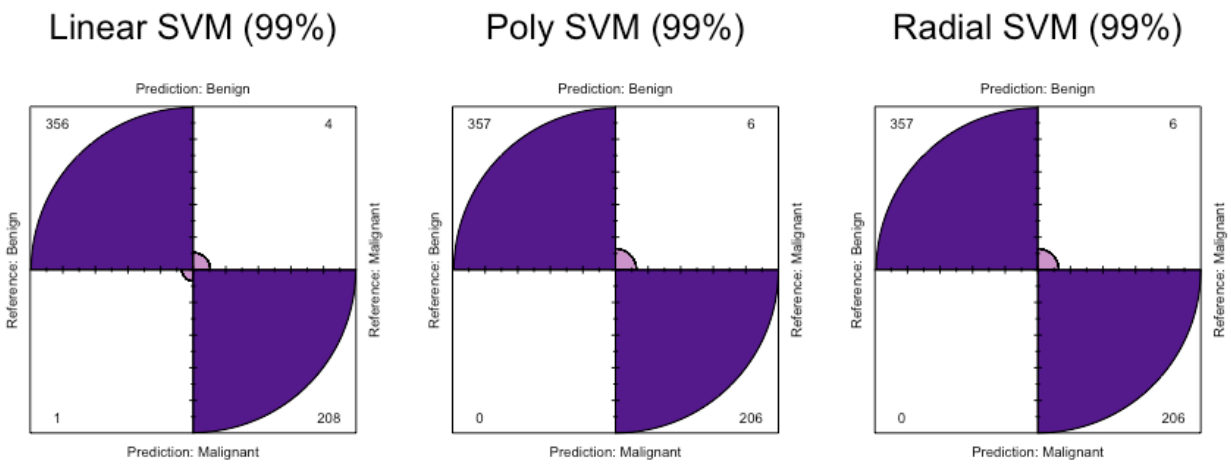


All four models performed well but the models with 30 predictors had fewer False Negatives. The full set with 30 predictors is the best KNN model.

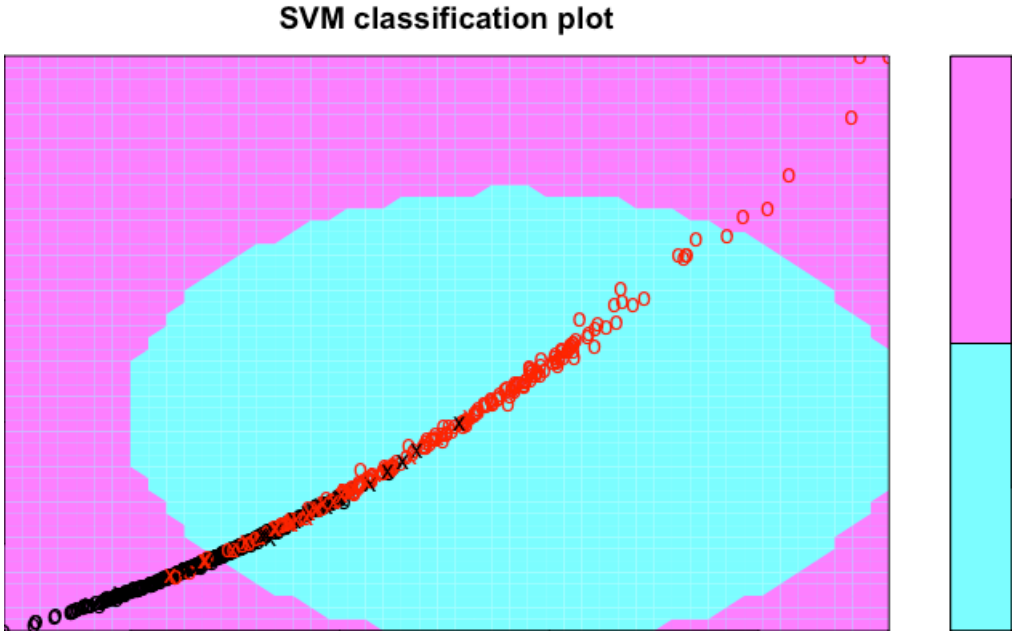
Support Vector Machines

```
256  ### SVM
257  # library(e1071)
258  # 30 predictors 4 neighbors 80/20 split
259  # Initialize Grid Values for CV:
260  gamma <- seq(0,0.1,0.005)
261  cost <- 2^(0:5)
262  deg <- c(2:4)
263  parms <- expand.grid(cost=cost, gamma=gamma)
264
265  set.seed(1)
266  # Find the best model:
267  svm.linear.tune.30 <- tune(svm, diagnosis~., data=train,
268                             kernel = "linear",
269                             ranges = list(cost = cost))
270  summary(svm.linear.tune.30)
271  svm.linear.tune.30.predict <- predict(svm.linear.tune.30$best.model, train)
272  table(pred = svm.linear.tune.30.predict, true = train$diagnosis)
273  mean(svm.linear.tune.30.predict != train$diagnosis)
274  # Create the best model
275  svm.linear.full <- svm(diagnosis~., data=wbcd,
276                         cost=svm.linear.tune.30$best.model$cost,
277                         kernel = "linear")
278  svm.linear.predict <- predict(svm.linear.full, wbcd[,-1])
279  # Create a Confusion Matrix
280  svm.linear.cm <- confusionMatrix(svm.linear.predict, wbcd$diagnosis)
```

```
282  set.seed(1)
283  svm.poly.tune.30 <- tune(svm, diagnosis~., data=train,
284                           kernel = "polynomial",
285                           ranges = list(cost = cost, degree = deg))
286  summary(svm.poly.tune.30)
287
288  set.seed(1)
289  svm.radial.tune.30 <- tune(svm, diagnosis~., data=train,
290                             kernel = "radial",
291                             ranges = list(cost = cost, gamma = gamma))
```



Of the three SVM models we can see that the Radial and Polynomial model worked better than the linear model. Therefore the optimal Radial SVM was made with a Cost $= 4$ and $\gamma = 0.005$.



Radial SVM plotted on the Area by Radius Plane.

K Means Clustering

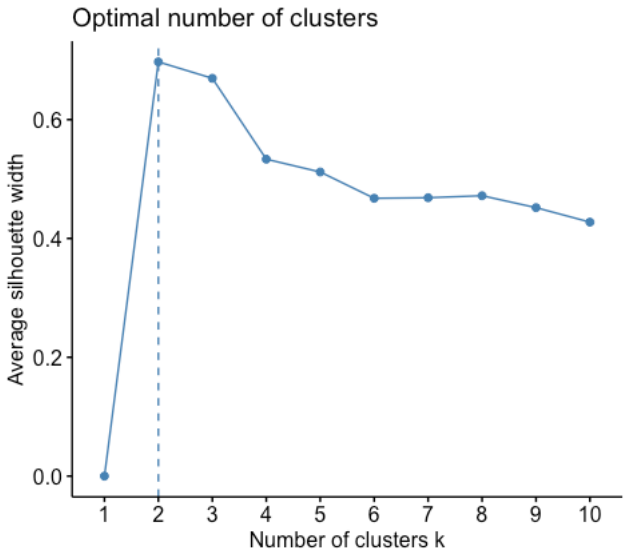
```
130  #### KMeans
131  #library(factoextra)
132  # Create a dataset of only the labels
133  wbcd.Labels = wbcd[1]
134  # Remove the labels (for clustering)
135  wbcd.30.cluster = wbcd[2:31]
136  wbcd.10.cluster = wbcd10[2:11]
137  # Scale the Data
138  wbcd.30.scale.cluster <- scale(wbcd.30.cluster)
139  wbcd.10.scale.cluster <- scale(wbcd.10.cluster)
140  ## WSS on 30 Predictors & Raw Data
141  set.seed(1)
142  fviz_nbclust(wbcd.30.cluster, kmeans,
143               nstart = 50,
144               method = "wss")
145  # Gap Statistic on 30 Predictors & Raw Data
146  set.seed(1)
147  fviz_nbclust(wbcd.30.cluster, kmeans,
148               nstart = 50,
149               nboot =20,
150               method = "gap_stat")
151  # Average Silhouette Width on 30 Predictors & Raw Data
152  set.seed(1)
153  fviz_nbclust(wbcd.30.cluster, kmeans,
154               nstart = 50,
155               nboot =20,
156               method = "silhouette")
```

Source Code to Generate Clusters

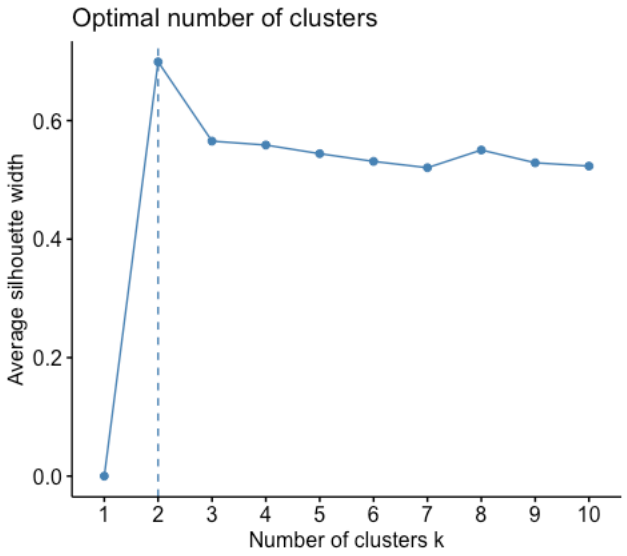
```
158  ## WSS on 30 predictors & Scaled
159  set.seed(1)
160  fviz_nbclust(wbcd.30.scale.cluster, kmeans,
161               nstart = 50,
162               method = "wss")
163  # WSS on 30 predictors & Scaled
164  set.seed(1)
165  fviz_nbclust(wbcd.30.scale.cluster, kmeans,
166               nstart = 50,
167               nboot =20,
168               method = "gap_stat")
169  # Average Silhouette Width on 30 predictors & Scaled
170  set.seed(1)
171  fviz_nbclust(wbcd.30.scale.cluster, kmeans,
172               nstart = 50,
173               nboot =20,
174               method = "silhouette")
```

Source Code to Obtain Silhouette Plots

Average Silhouette Width on Unscaled Data:



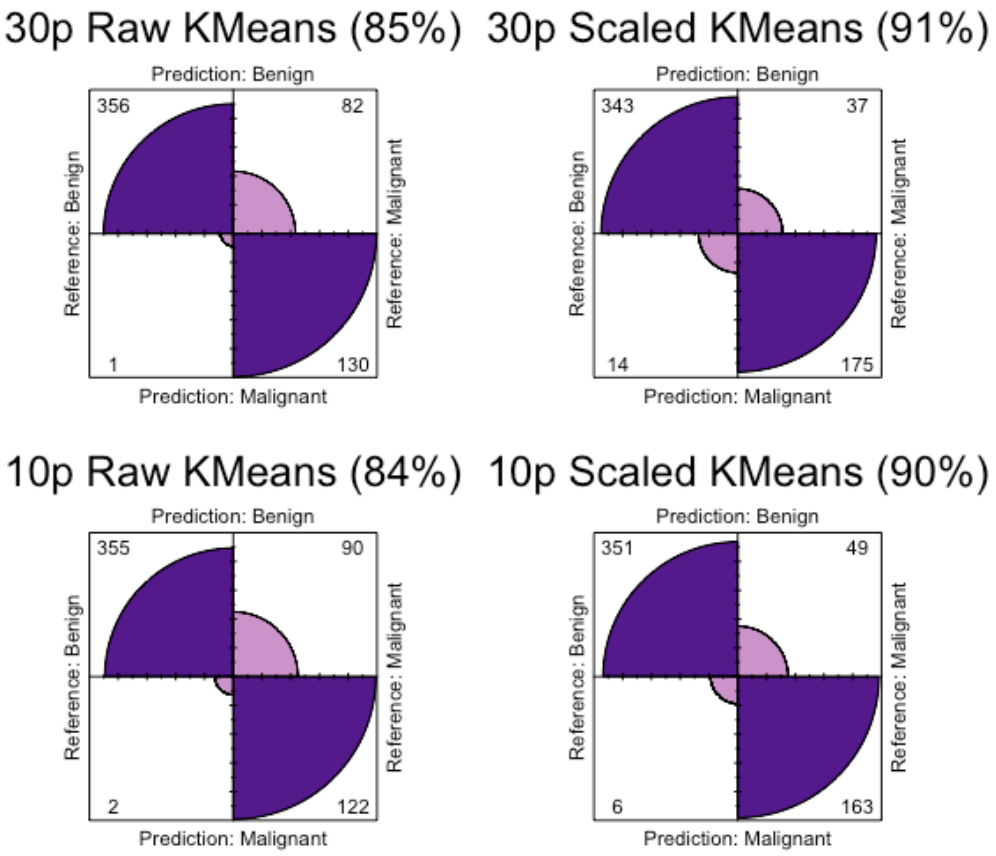
30 Predictors

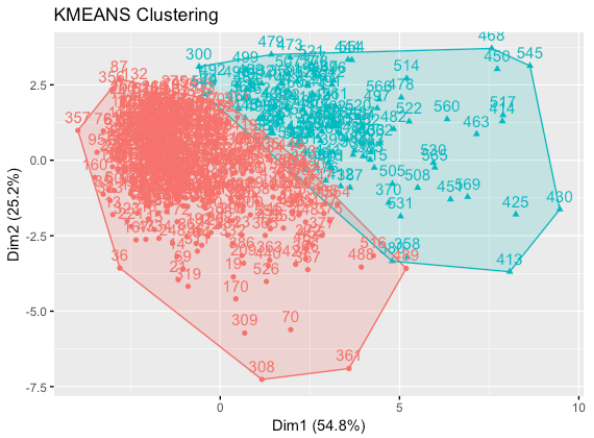


10 Predictors

We can compare the cluster assignments by finding the majority of the class labels of each cluster. Then asiigning every observation in that cluster as that label. Finally I comparing the cluster label with the actual label to obtain a confusion matrix.

```
### Creating Confusion Matrices for KMeans:
# Create optimal KMeans models:
# Optimal KMeans with 30 predictors & Raw Data
set.seed(1)
km.30 <- eclust(wbcd.30.cluster, FUNcluster = "kmeans", k=2, nstart=50)
# Obtain the cluster assignments from your object
km.30.raw.assign <- km.30$cluster
# Convert cluster assignments from numerical to it's categorical equivalent
km.30.raw.test <- factor(ifelse(km.30.raw.assign ==1,"Benign","Malignant"))
# Create a confusion Matrix
km.30.raw.cm <- confusionMatrix(km.30.raw.test, wbcd10$diagnosis)
km.30.raw.caption <- paste("30p Raw KMeans (",round(km.30.raw.cm$overall[1]*100),"%)",sep="")
```



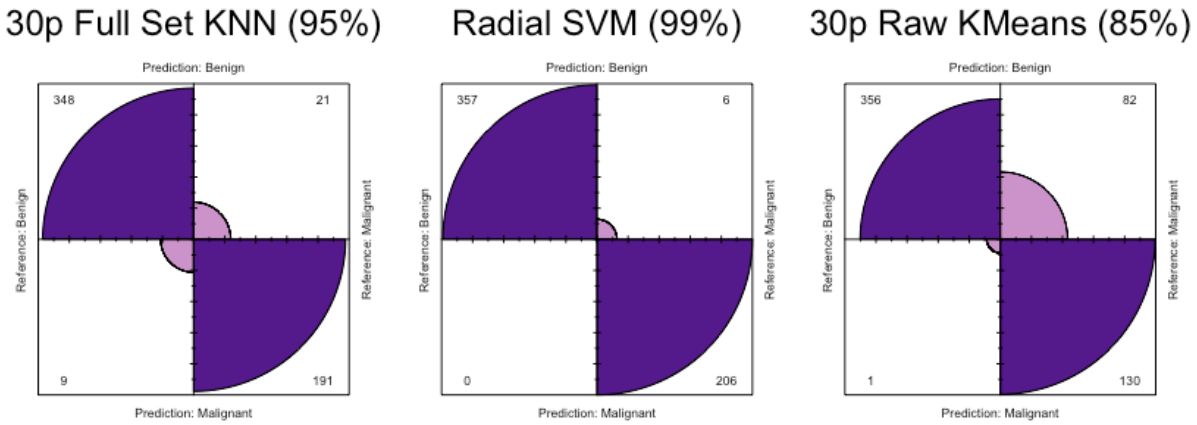


Raw Data

Scaled Data

The Scaled models worked better but had much more False Negatives. Additionally I can see a slight imporvement with the 30 predictor models compared to the 10 predictor models.

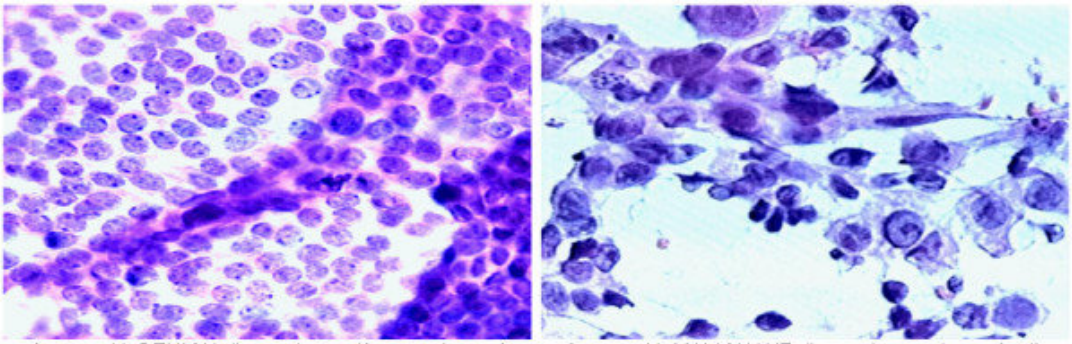
4. Analysis:




```
> mean(knn.pre.30.full != wbcd[,1])
[1] 0.05272408
> mean(svm.poly.tune.30.predict != train$diagnosis)
[1] 0.01098901
> mean(km.30.raw.test!= wbcd10$diagnosis)
[1] 0.1458699
```

The Radial SVM Model has the best Training Error.

5. Conclusions:



Smear with BENIGN diagnosis – uniform nucleus of cells, symmetrical, homogeneous, with areas within normal size

Smear with MALIGNANT diagnosis – nucleus of cells without uniformity, asymmetrical, not homogeneous (multiple sizes) and with areas above normal size

Source: Wikipedia Commons

From a purely visual standpoint, the malignant cells are more ellipsoidal and the benign cells are more spherical. Since the malignant tumor cells have a high surface area/mass ratio they were naturally plotted in the same location in 30-dimensional space. The correlation between area, perimeter, coastline approximation, and radius would make it easier to distinguish between malignant cells or benign cells.

KNN naturally showed weaker performance in 30-dimensional space compared to the SVM model. More importantly, there are no huge differences between the models with 30 predictors or 10 predictors. Perhaps, the KNN models already fell prey to the "curse of dimensionality" with 10-dimensional space. If tasked to continue this project further, I could test if the KNN models improve over (or even outperform) SVM if I was to add many patients to the data set.

The KMeans clustering methods, although performed well, they did not match the worst performing supervised models. If I was tasked with a different goal then KMeans could help us see how the observations intrinsically coalesce into clusters. All of the clustering validation tests reinforced the fact that the data had two natural clusters.

Regardless of the model used - I am very pleased with the results of all the models. All three models performed well.

6. References:

[1] Henderson LM, Hubbard RA, Sprague BL, Zhu W, and Kerlikowske K: "Increased Risk of Developing Breast Cancer after a False-Positive Screening Mammogram", Cancer Epidemiol Biomarkers Prev, December 1 2015 (24) (12) 1882-1889; DOI: 10.1158/1055-9965.EPI-15-0623 link

[2] Hubbard RA, Kerlikowske K, Flowers CI, Yankaskas BC, Zhu W, Miglioretti DL. Cumulative probability of false-positive recall or biopsy recommendation after 10 years of screening mammography: a cohort study. Ann Intern Med. 155(8):481-92, 2011.

[3]Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. link.

[4]K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34 link.