Computer Vision & ML - Skin Cancer Diagnostics

Table of Contents

1. Project Description and Summary (1 page)

- **Goal** Goal of the project is to:
  - download and process annotated images of skin moles in JPEG format, half of which are benign and the remaining malignant.
  - construct 3 different classification models for identifying malignant moles based on the pixels (RGB colors) of these images and demonstrate the model accuracies.
  - define new data processing/feature engineering approaches that could improve the interpretability of the machine learning classification models.
  - apply 2 different classification models on the feature engineered skin image data and demonstrate that the model accuracies have increased.
- **Approach**
  - Load the images using EBImage library. EBImage provides general purpose functionality for image processing,analysis and facilitates statistical modeling, machine learning and visualization with image data.
  - For Part 1, considering the fact that all the images are not of the same size,resize all the images to 256 X 256 X 3 (3 channels - RGB) and vectorize the image arrays for further processing.
  - For Part 1, define 3 pixel based machine learning models based on the following algorithms - Gradient Boosting Machines(GBM), K-Nearest Neighbor(KNN), Naive-Bayes (NB) and calculate the accuracies on Train and Test data (70 - 30 split)
  - For Part 2, perform the following additional image processing and feature engineering steps using EBImage package before vectorizing the image data and using it for classification.
    1. image brightness,contrast adjustment
    2. adaptive image thresholding
    3. image resizing to 256 X 256 X 3
    4. image cropping (to size 200 X 200 X 3) to remove unwanted image borders
    5. Principal Component Analysis (PCA) to select the most prominant features for further machine learning
    6. Compute additional shape features using EBImage Package
    7. Use computed shape features to derive border additional features - irregularity index and geometrical asymmetry measure
  - For Part 2, fit 2 machine learning models - K-Nearest Neighbor(KNN), Gradient Boosting Machines(GBM) on the feature engineered data and calculate the accuracies on Train and Test data (70 - 30 split)

- **Results**

Results obtained show that additional feature engineering, especially those engineered based on clinical papers and domain knowledge of the skin cancer definitely contribute towards increasing the classification accuracy when it comes to identifying malignant skin lesions versus benign ones, thus aiding more in the detection of skin cancer.

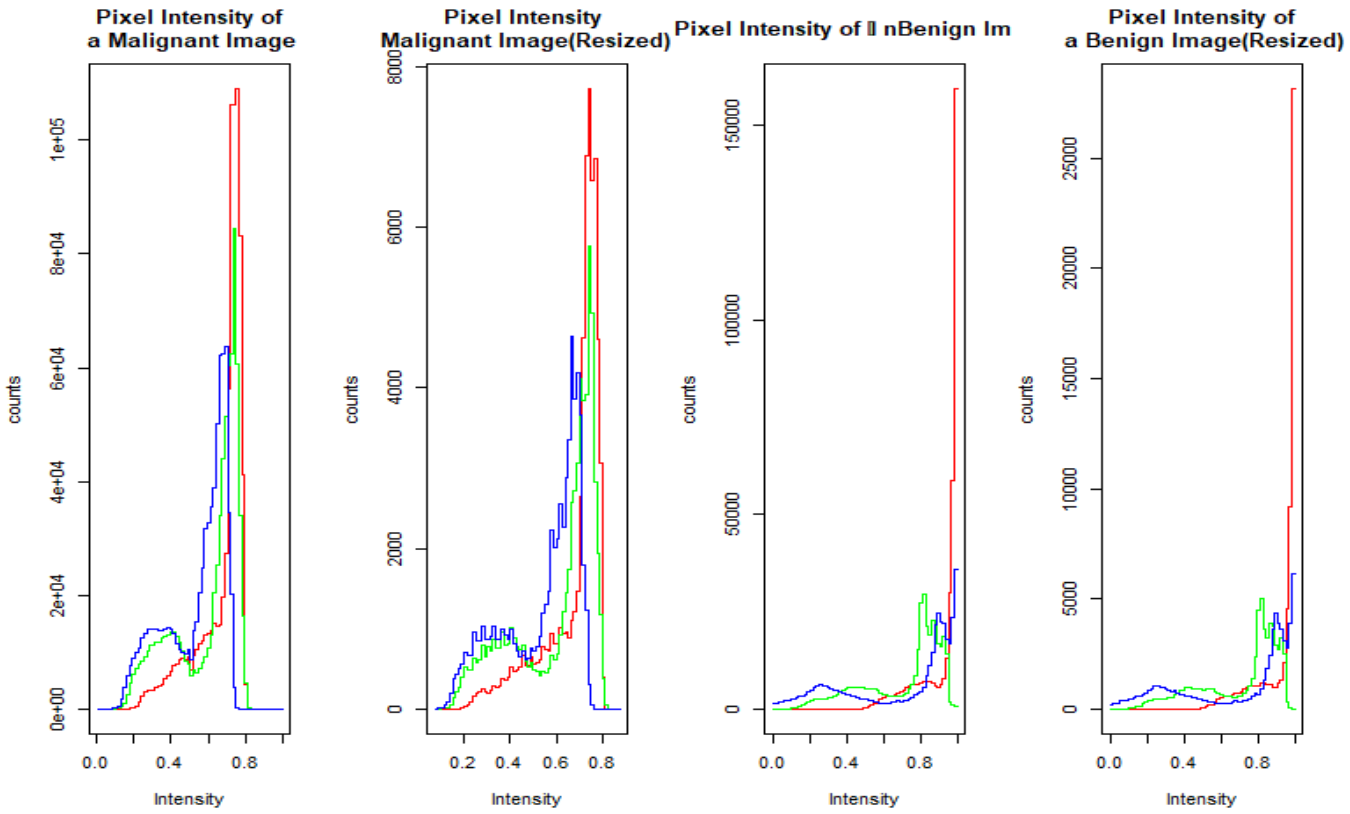
The below details based on which this conclusion has been made are also captured in this report.

  - Classification Results and Metrices for Part 1 - All 3 Models
  - Classification Results and Metrices for Part 2 - Both the Models
  - Comparison of Model classification results for Part 1 and Part 2

2. Data Processing

Data Processing (for Part 1) steps: Use R's EBImage library to load and resize all the images to 256 X 256 X 3 (3 channels - RGB) and vectorize the image arrays for further processing as shown below

```
#####################################################################
#Benign Image Processing Using EBImage library(for Part 1)
#####################################################################
library(EBImage) #image processing library
ben.files <- list.files("Data/benign") #list of all benign image files
mal.files <- list.files("Data/malignant") #list of all malignant image files
ben.df <- data.frame()
for (i in 1:150) {
  ben.path <- paste("Data/benign/", ben.files[i], sep="")
  b.img <- readImage(ben.path)
  rz.b.img <-  resize(b.img, w=256, h=256)# #resize each image
  rz.bimg.vec <- as.vector(rz.b.img) #vectorize
  labelled.bimg.vec <- c(0, rz.bimg.vec) #add a new column with '0' as label
  #append to the dataframe
  ben.df <- rbind(ben.df, labelled.bimg.vec) }
#name all the columns in the dataframe
names(ben.df) <- c("label", 1:(ncol(ben.df)-1))
```



3. Classification Models Based on Pixels

3.1 Model 1

- **Model Details & Tuning**
  First classifier model used for identifying malignant skin cancer is Naïve Bayes classifier which is a simple probabilistic classifier based on Bayes theorem which is based on concept of variable independence. Trainer from caret package is used; parameter 'nb' indicates to use Naive Bayes. The trainControl part tells the trainer to use cross-validataion ('cv') with 10 folds. xTrain 1 is the train data having pixel values and yTrain1 has labels indicating '0' for benign and '1' for malignant images. Model will be fitted on xTest1 data and predictions will be done classsifying each image as benign (0) or malignant(1). Details of the model and prediction statistics are given below:

```
#####################################################################
#Part 1 - Model 1: Naive Bayes Classifier Details & Tuning
#####################################################################
  library(e1071)
  library(caret)
  set.seed(45)
  #fitting the NB classifier on train data
  q1nb.fit = train(xTrain1,
as.factor(yTrain1),'nb',trControl=trainControl(method='cv',number=5))
  q1nb.fit

## Naive Bayes
##
## 210 samples
## 300 predictors
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 168, 167, 168, 169, 168
```

```
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.7331416  0.4616970
##    TRUE      0.7139832  0.4260055
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning parameter 'adjust'
##  was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust = 1.
```

- **Model Validation**

```
## Part 1 NB - TEST DATA PREDICTION STATISTICS

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29 12
##          1 20 29
##
##                Accuracy : 0.6444
##                  95% CI : (0.5365, 0.7426)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 0.03514
##
##                   Kappa : 0.2945
##
##  Mcnemar's Test P-Value : 0.21592
##
##             Sensitivity : 0.5918
##             Specificity : 0.7073
##          Pos Pred Value : 0.7073
##          Neg Pred Value : 0.5918
##              Prevalence : 0.5444
##          Detection Rate : 0.3222
##    Detection Prevalence : 0.4556
##       Balanced Accuracy : 0.6496
```

3.2 Model 2

- **Model Details and Tuning**

Second classifier used for identifying malignant skin cancer is K-Nearest Neighbor(KNN) classifier which is a non-parametric method. Here knn from 'class' package is used and the output is a class membership - benign (0) or malignant (1). An object is classified based on vote from its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

A common rule of thumb while tuning parameter 'k' is to use a rounded value of the square root of total number of elements in the dataset on which the classifier will be fit. In this case that would be square root of 210 = 14.4 (hence k = 14 is chosen). This choice of k proved to be the best in terms of accuracy based on trial on multiple k values.

Details of the model and prediction statistics are given below:

```
#########################################################################
#Part 1 - Model 2: K Nearest Neighbor (KNN) - Model Details & Tuning
#########################################################################
library(class)
#fit KNN model on train data and predict ; number of classes = 2
```

```
set.seed(10)
q1knn.yPredTrain <- knn(train = xTrain1, test = xTrain1, cl=yTrain1 , k=14)
q1knn.yPredTest <- knn(train = xTrain1, test = xTest1, cl=yTrain1 , k=14)
```

- **Model Validation**

```
## Part 1 KNN - TEST DATA PREDICTION STATISTICS

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 39 23
##          1 10 18
##
##                Accuracy : 0.6333
##                  95% CI : (0.5251, 0.7325)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 0.05540
##
##                   Kappa : 0.2412
##
##  Mcnemar's Test P-Value : 0.03671
##
##             Sensitivity : 0.7959
##             Specificity : 0.4390
##          Pos Pred Value : 0.6290
##          Neg Pred Value : 0.6429
##              Prevalence : 0.5444
##          Detection Rate : 0.4333
##    Detection Prevalence : 0.6889
##       Balanced Accuracy : 0.6175
```

3.3 Model 3

- **Model Details and Tuning**

Third classifier used for identifying malignant skin cancer is Gradient Boosting classifier (GBM) which is an ensemble method. The output is a class membership - benign (0) or malignant(1). Gradient boosting classifiers produces a prediction model in the form of an ensemble of weak prediction models called stumps.

The gbm() function from 'gbm' package has been used in the code below. This uses a learning rate (shrinkage) of 0.001,default number of trees of 100 with default depth of each tree (interaction.depth) 1, which means we are ensembling a bunch of stumps. A cross validation using cv.folds is used to to perform a 10 fold cross validation.

Details of the model and prediction statistics are given below:

```
################################################################################
###
#Part 1 - Model 3: GBM (Gradient Boosting Machines) Classifier - Model Details& Tuning
################################################################################
###
library(caret)
library(gbm)
set.seed(0)
#fitting the NB classifier on train data
q1gb.fit =
train(xTrain1,as.factor(yTrain1),'gbm',trControl=trainControl(method='cv',number=10))
q1gb.fit
```

- **Model Validation**

```
## Part 1 GBM - TEST DATA PREDICTION STATISTICS

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 36 16
##          1 13 25
##
##                Accuracy : 0.6778
##                  95% CI : (0.571, 0.7725)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 0.006912
##
##                   Kappa : 0.3465
##
##  Mcnemar's Test P-Value : 0.710347
##
##             Sensitivity : 0.7347
##             Specificity : 0.6098
##          Pos Pred Value : 0.6923
##          Neg Pred Value : 0.6579
##              Prevalence : 0.5444
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.5778
##       Balanced Accuracy : 0.6722
```

4. Literature Review for Part 2

Below is a summary of literature review that has motivated the feature engineering for Part 2:

- **1.American Cancer Society** (http://www.cancer.org/cancer/melanoma-skin-cancer.html)

  American Cancer Society has defined the ABCDE rule as a guide to check if a particular skin lesion is benign (non-cancerous) or malignant (melanoma or skin cancer).

  Melanoma spots often seem to have one or more of the following ABCDE features:

  ● A is for Asymmetry: One half of a mole or birthmark does not match the other.

  ● B is for Border:The edges are irregular, ragged, notched, or blurred.

  ● C is for Color:The color is not the same all over and may include different shades of brown or black, or sometimes with patches of pink, red, white, or blue.

  ● D is for Diameter:The spot is larger than 6 millimeters across (about ¼ inch – the size of a pencil eraser), although melanomas can sometimes be smaller than this.

  ● E is for Evolving: The mole is changing in size, shape, or color.

- **2.NIH Clinical Paper** - Title: 'A systematic heuristic approach for feature selection for melanoma discrimination using clinical images' (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193077/)

  Of the ABCDE clinical features mentioned above, the above clinical paper discusses about how the shape features of the lesions identified from clinical images, particularly 'A' and 'B' features - asymmetry and border irregularity - can be used to derive features that could be used to classify skin cancer images.

The paper proposes the use of below computed shape features based on quantifying border irregularity and lesion asymmetry.

(a) **Border irregularity index** In order to quantify border irregularity, an irregularity index is computed for each lesion. The irregularity index is given by the formula - I = (ab / 2pi((a2)+(b2)) (P^2/A) where a and b are the lengths of the major and minor axes of the best-fit ellipse, respectively, P is the perimeter of the lesion border, and A is the area of the lesion. This could be used a a new feature by the classification model for skin cancer identification from images.

- **3.Journal of Artificial Intelligence** Article 'Asymmetry Analysis of Malignant Melanoma Using Image Processing: A Survey' from (https://scialert.net/fulltextmobile/?doi=jai.2014.45.53#e5)

  The above article was referred which proposed the use of geometrical symmetry as a feature to identify skin cancer lesions.

  (b) **Geometrical asymmetry** Geometric asymmetry can be found by dividing the lesion into 2 parts by straight line that passes through the center of mass, after that a comparison is made between the 2 parts by calculating the distance present between the size functions. This size functions distance also determines the qualitative asymmetry.

  Based on this a geometrical asymmetry feature can be calculated as 'difference between minimum radius of the lesion and maximum raduis of the lesion' and this can also be used as a new feature by the calssification model for skin cancer identification from images.

5. Feature Engineering for Part 2

Below are the main feature engineering steps that were done as part of Part 2

- Calculation of **border irregularity index** feature. This value that quantifies the border irregularity of the lesion was calculated using the formula - I = (ab / 2*pi(a^2 + b^2)) (P^2/A) - based on the medical literature mentioned above. The axes values , area and perimeter values were calculated using the EBImage package 'computeFeatures.shape()' which returns the following shape related features of the lesion - A = s.area, P = s.perimeter, b = s.radius.min, a = s.radius.max

- Calculation of **geometrical asymmetry measure** feature. This value that quantifies the asymmetry of the lesion was calculated using the formula - GA = (max raduis - min radius) - based on the medical literature mentioned above. The radius values were calculated using the EBImage package 'computeFeatures.shape()' which returns the following shape related features of the lesion - min radius = s.radius.min, max radius = s.radius.max

- The above two derived features, along with the **area** and **perimeter** features returned by the EBImage function computeFeatures.shape() were used for the calssification process

- In addition to the above, the actual image itself was subjected to the following **image enhancement** steps using EBImage package making it easier to identify key features of the lesion when extracted , compared to the surrounding skin area.

```
1. Brightness and Contrast adjustment - Adjusting brightness and contrast makes
sure that the lesion is clearly visible in the image compared to the surroundings.
Brightness and contrat can be increased by adding or multiplying a positive value
to the image respectively - (img + 0.6) or (img * 2)

2. Adaptive Image Thresholding - EBImage's threshold() function separate "object"
or foreground pixels from background pixels to aid in image processing.It also
nullifies the affect of uneven illumination or by stray signal from nearby bright
objects.

3. Resizing - EBImage function resize() performs resizing by scaling the image x
to the desired dimensions.This was important as not all of our images were of the
```

exact size

4. **Crop Images** - Images were also uniformly cropped to reduce the noise line presence of borders in some cases.

5. **PCA** - Principal Component Analysis or PCA , a dimensionality reduction procedure was also done on these images post enhancement and vectorization to make sure that only the most prominant of almost the 120k pixels were selected to be used for the classification model.

```
##############################################################################
#Raw Image Processing and Feature Engineering Using EBImage library(for Part 2)
##############################################################################
# - Lesion shape feature extraction & Feature Engineering - benign
library(EBImage) #image processing library
#Load benign images
ben.df <- data.frame()
for (i in 1:150) {
  ben.path <- paste("Data/benign/", ben.files[i], sep="")
  b.img <- readImage(ben.path)
  gray.b.img<-channel(b.img,"gray")
  b.img.ft = computeFeatures.shape(x=bwlabel(gray.b.img)) # access shape features
  #Border irregularity index - b.irr.f
  b.area = b.img.ft [1]
  A = b.area #area
  b.perimeter = b.img.ft [2]
  P = b.perimeter #perimeter
  b.major.ell.axlen = b.img.ft [6] * 2
  a = b.major.ell.axlen  #major axis length
  b.minor.ell.axlen = b.img.ft [5] * 2
  b = b.minor.ell.axlen #minor axis length
  b.num = (( a * b ) / (2*3.14*(a^2 + b^ 2)))
  b.den = (P^2 / A)
  b.irr.f = b.num/b.den
  #geometrical asymmetry measure
  g.asy.m = (b.img.ft [6] - b.img.ft [5])
  #combining all shape features
  b.img.f = cbind(A,P,g.asy.m ,b.irr.f)
  #vectorize
  rz.bimg.vec <- as.vector(b.img.f)
  #add a new column with '0' as label
  labelled.bimg.vec <- c(0, rz.bimg.vec)
  #append to the dataframe
  ben.df <- rbind(ben.df, labelled.bimg.vec)
}
#name all the columns in the dataframe
names(ben.df) <- c("label", 1:(ncol(ben.df)-1))
```

6. 2 CLASSIFICATION MODELS for Part 2

6.1 Model 1

- **Model Details and Tuning**

Model 2 for Part 2 is a Gradient Boosting classifier (GBM) classifier.Parameters and settings chosen are same as that of GBM in Part 1, for comparison purpose.

Details of the model and prediction statistics are given below:

```
#####################################################################
#Part 2 - Model 1: GBM (Gradient Boosting Machines) Classifier
```

```
#- Details and Tuning
#######################################################################
library(caret)
library(gbm)
set.seed(0)
#fitting the GB classifier on train data
q2gb.fit =
train(xTrain2,as.factor(yTrain2),'gbm',trControl=trainControl(method='cv',number=10))
q1gb.fit
```

- **Model Validation**

```
## Part 2 GBM - TEST DATA PREDICTION STATISTICS

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 36 13
##          1 13 28
##
##                Accuracy : 0.7111
##                  95% CI : (0.606, 0.8018)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 0.0008961
##
##                   Kappa : 0.4176
##
##  Mcnemar's Test P-Value : 1.0000000
##
##             Sensitivity : 0.7347
##             Specificity : 0.6829
##          Pos Pred Value : 0.7347
##          Neg Pred Value : 0.6829
##              Prevalence : 0.5444
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.5444
##       Balanced Accuracy : 0.7088
```

6.2 Model 2

- **Model Details and Tuning** Second classifier used for Part 2 is K-Nearest Neighbor(KNN) classifier from 'class' package is used and the output is a class membership - benign (0) or malignant(1). Square root of data length 210 = 14.4 (hence k = 14 is chosen).

Parameters and settings chosen are same as that of KNN in Part 1, for comparison purpose.

Details of the model and prediction statistics are given below:

```
#####################################################################
#Part 2 - Model 2: K Nearest Neighbor (KNN) Details and Tuning
#####################################################################
library(class)
set.seed(10)
#fit KNN model on train data and predict ; number of classes = 2
q2knn.yPredTrain <- knn(train = xTrain2, test = xTrain2, cl=yTrain2 , k=16)
q2knn.yPredTest <- knn(train = xTrain2, test = xTest2, cl=yTrain2 , k=16)
```

- **Model Validation**

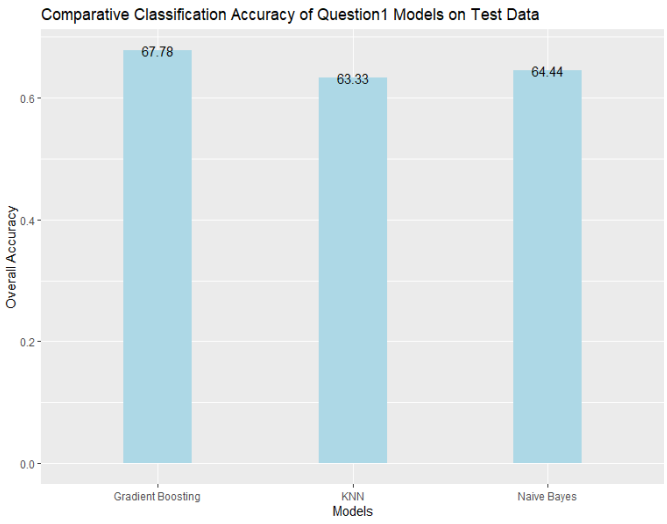
```
## Part 2 KNN - TEST DATA PREDICTION STATISTICS
```

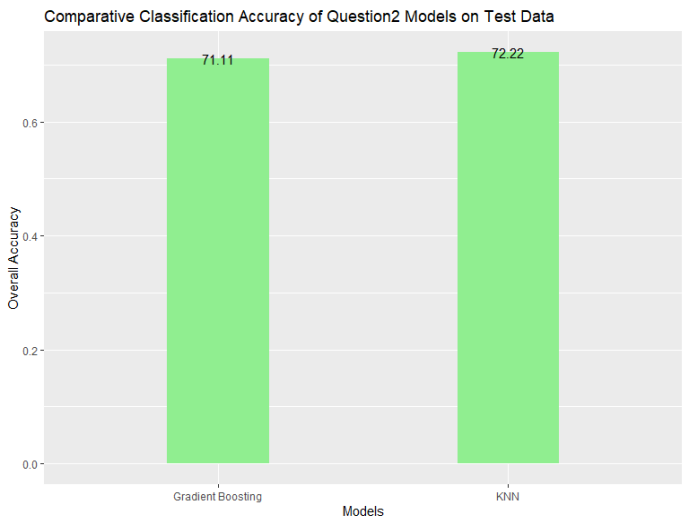
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 39 15
##          1 10 26
##
##                Accuracy : 0.7222
##                  95% CI : (0.6178, 0.8115)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 0.0004108
##
##                   Kappa : 0.4344
##
##  Mcnemar's Test P-Value : 0.4237108
##
##             Sensitivity : 0.7959
##             Specificity : 0.6341
##          Pos Pred Value : 0.7222
##          Neg Pred Value : 0.7222
##              Prevalence : 0.5444
##          Detection Rate : 0.4333
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.7150
```

6.3 Model Results for Part 1 and Part 2 - Interpretation

Below plots show the prediction accuracies of 3 classification models from Part 1 where only pixel data are used as features and for classification and 2 classification models from Part 2 where engineered features based on inputs from medical literature for skin cancer are used for classification.

KNN and GBM algorithms are used for classifying benign and malignant skin lesions, under both Part 1 and Part 2 with the same parameters. Based on the accuracies(plotted below) , sensitivity and specificity (listed in the above sections) it is clear that **feature engineering has improved the accuracy of Part 2 models**).





The increase in accuracy can obviously attributed to the fact that additional features like Area, Perimeter, Border Asymmetry index, Geometrical asymmetry etc that relate to the 'ABCDE' features for identifying skin cancer has been used in training Part 2 models where as Part 1 models were just trained on pixel data alone and nothing else. Incorporating domain specific features have indeed increased the accuracy.