STAT 542 / CS 598: Project 2

Fall 2019, by Prathamesh(satpute3), Vivek(vivekg3) and Athul(as81)

Due: Monday, Dec 16 by 11:59 PM Pacific Time

[10 Points, half a page] Project description and summary. This part should summerise your goal, your approach, and your results.

half page description goes here

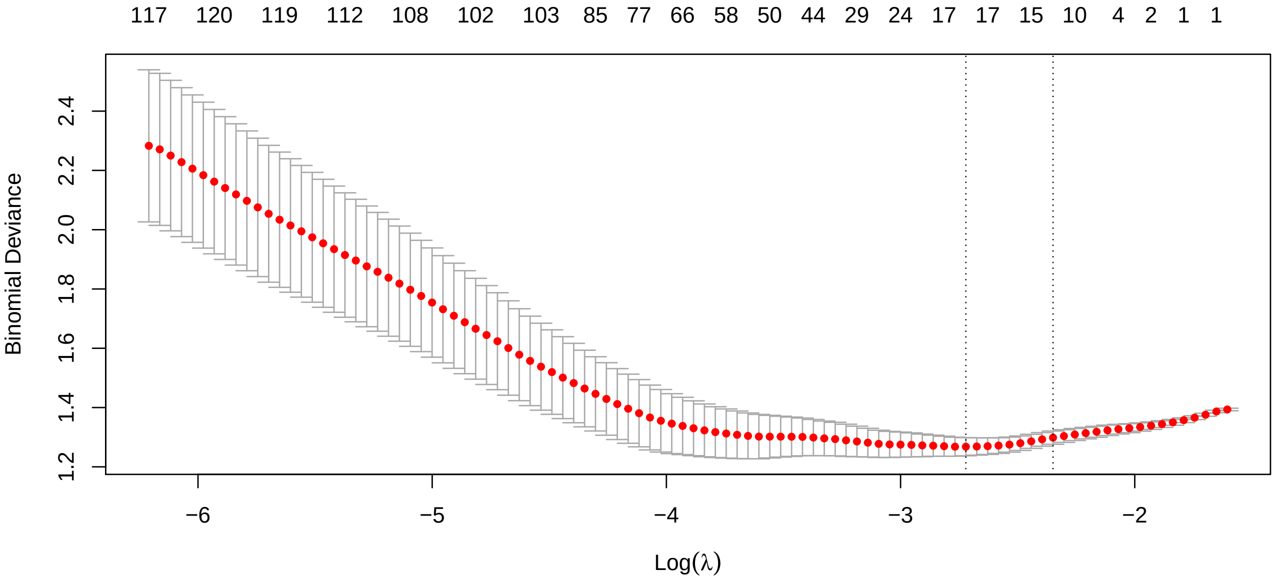
[5 Points, half a page] Data processing for Question 1. Describe how you process the data so that it can be analyzed to answer question 1.

half page description goes here

[30 Points, within 5 pages] Classification models based on pixels.

```
  # kable(results, caption = "KNN")
```

```
# use parallel for performace
  registerDoMC(cores = 4)
  cv_glmnet_model <- cv.glmnet(train.data[, -1], train.data[, 1], parallel = TRUE, alpha=1, family="binomi
```



```
  best_lambda = cv_glmnet_model$lambda.1se
  # training with best lambda selected from the cv
  train_glmnet_model <- glmnet(train.data[, -1], train.data[, 1], lambda = best_lambda, alpha=1, family="b

  summary(train_glmnet_model)
```

Table 1: Lasso Regression Results

Best.lambda	Accuracy
0.0954646	0.6333333

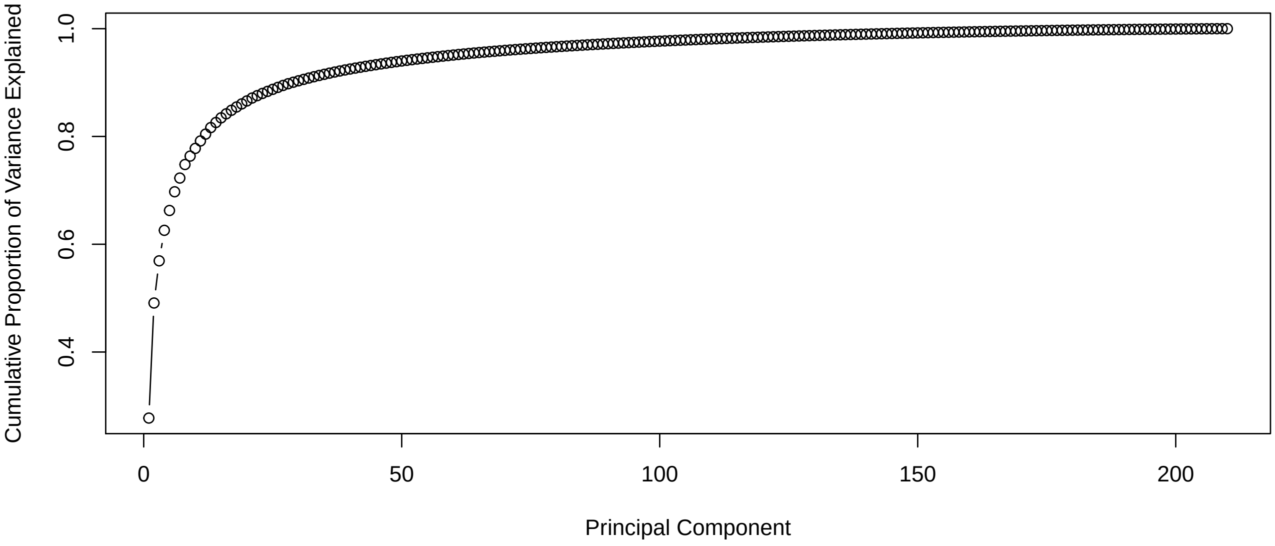
```
##            Length Class     Mode
## a0             1  -none-    numeric
## beta       30000  dgCMatrix S4
## df             1  -none-    numeric
## dim            2  -none-    numeric
## lambda         1  -none-    numeric
## dev.ratio      1  -none-    numeric
## nulldev        1  -none-    numeric
## npasses        1  -none-    numeric
## jerr           1  -none-    numeric
## offset         1  -none-    logical
## classnames     2  -none-    character
## call           6  -none-    call
## nobs           1  -none-    numeric
```

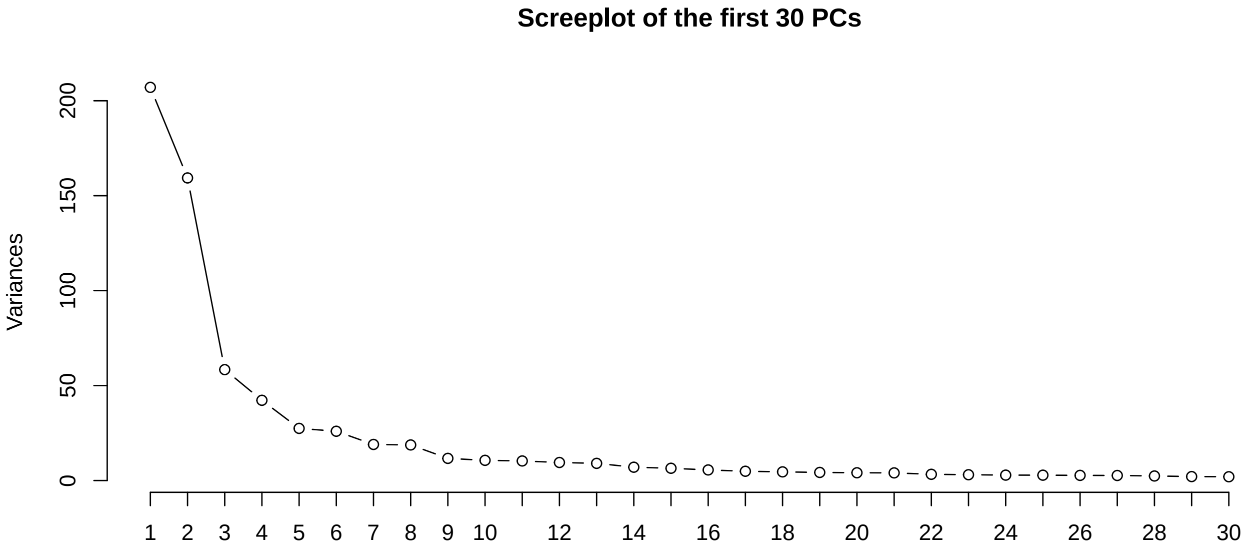
```
pred <- predict(train_glmnet_model, s = best_lambda, newx = test.data[, -1], type = "class")
accuracy = mean(test.data[, 1] == pred)

results = data.frame("Best lambda" = best_lambda, "Accuracy" = accuracy)
```

```
kable(results, caption = "Lasso Regression Results")
```

PCA is an excellent choice when it comes to images because inherently due to its natur, there is spatial corellation among pixel. Instead of using all the pixes, we can signifcantly reduce the number of features which encompass most of the variation. We choose 27 number of components which accounts for over 90% of the variation.




Screeplot of the first 30 PCs
Variances
0
50
100
150
200
1
2
3
4
5
6
7
8
9
10
12
14
16
18
20
22
24
26
28
30

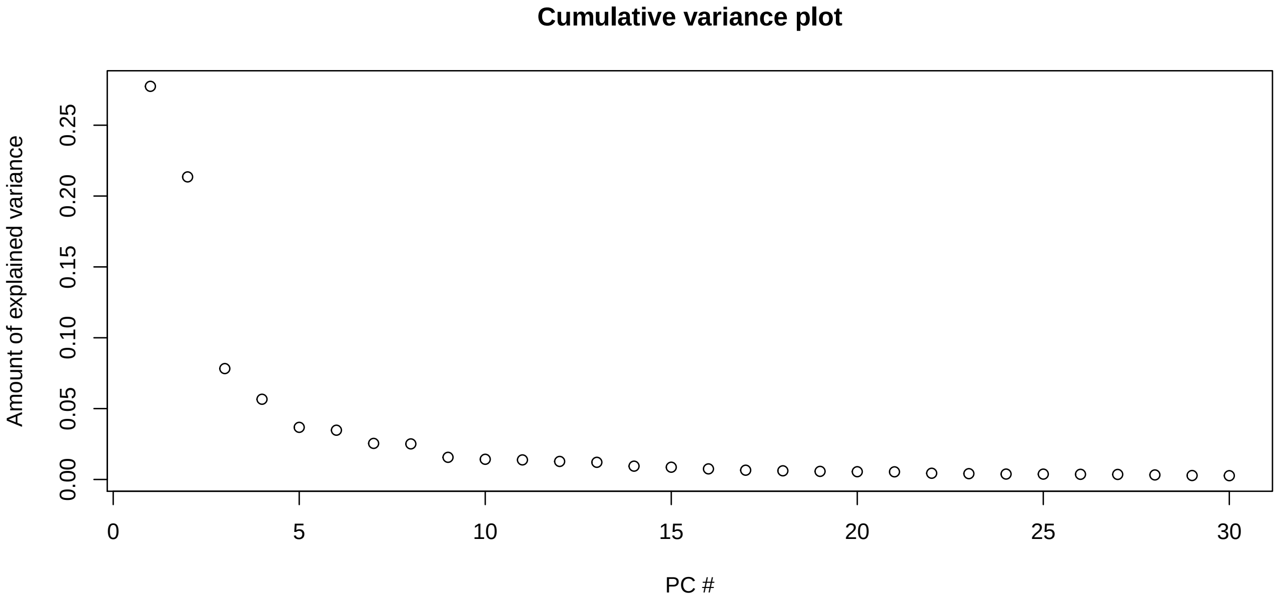
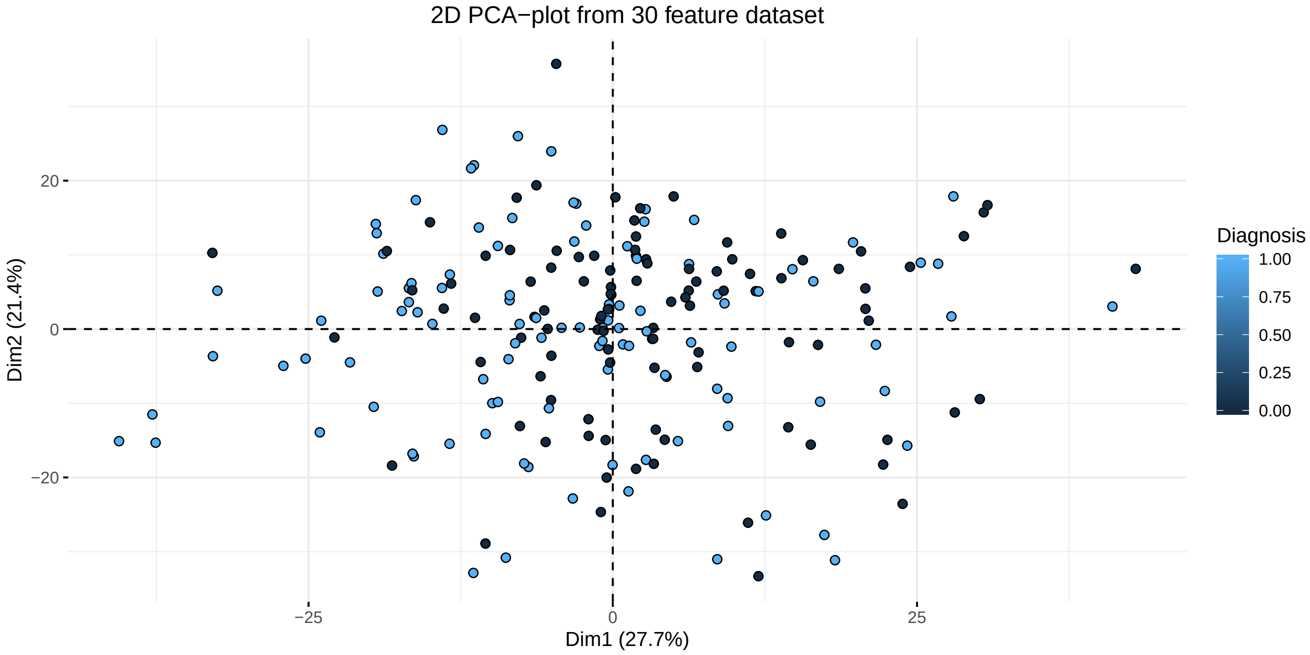

Cumulative variance plot
Amount of explained variance
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
15
20
25
30
PC #

Table 2: RandomForest Confusion Matrix

	0	1
0	140	10
1	10	140



We run a random forest through a grid search through of number of trees (1 to 10000) and nodesize(1 to 50). We find that our best model is with 1250 trees and a nodesize of 5. Our accuracy on the entire set is 93.33%

```
yhat.all <- predict(best_rfModel, all_data_x)
temp <- cbind(as.numeric(as.character(yhat.all)), (all_data_y))
all_error <- length(which(temp[, 1] != temp[, 2])) * 100 / nrow(all_data_x)
temp <- confusionMatrix(as.factor(yhat.all), as.factor(all_data_y))
kable(temp$table, caption="RandomForest Confusion Matrix")
```

[10 Points, 1 page] Literature review. You should search and read existing literature and summarize clinically relevant characteristics that could be used for skin cancer image diagnosis. There is no limitation on what type of literature you could use. However, the goal should be motivating your feature engineering approaches from a clinical and analytic point of view. Please give appropriate citations to the literature you read.

The American Academy of Dermatology Association (@aad) listes the ABCDE of detecting Melanoma. ABCDE is an acronym for Asymmetry, Border, Color, Diameter and Evolving respectively. Since in this project statement, are given a series of pictures over a lesion over a period of time, we will not be able to create a specific feature for Evolving. Hence, we concentrate on the remaining.
A great tools for us is EBImage which is package developed by @Pau2010 which provides us very usable functions for image processing tasks. We use various features of this package for our feature engineering. @RA2012 provides high level features one could use to help engineer our features. Their main work talks about the different way to quantize irregularities. They use a mix of both coarse and fine grain methods to achieve this. @SJ2015 Shivangi et al, also talks about a good pipeline to this. Their work diffs in that they introduce quantitative metrics to mark irregulaties in shape. They also take into account the size of the lesion. However to this accurately they must have ensured that the each image is taken from the same distance and focus. We do not have that information

regarding our dataset, so we must rely on ration. One of the interesting features proposed in using Circularity Index, it is the ration of (4piArea)/(Perimeter*Perimeter). This is a geat metric becaus this is scale invariant. Of course is @SJ2015 and @Pau@2010, they do concentrate a fair amount on preprocessing steps like illumination and segmentation.

[10 Points, 1 page] Feature engineering. Motivated by what you have read (or your understanding), process the data in a reasonable way such that the new variables are more intuitive to your collaborator/clinicians. You need to describe clearly what is your data processing criteria and how your variables are calculated.

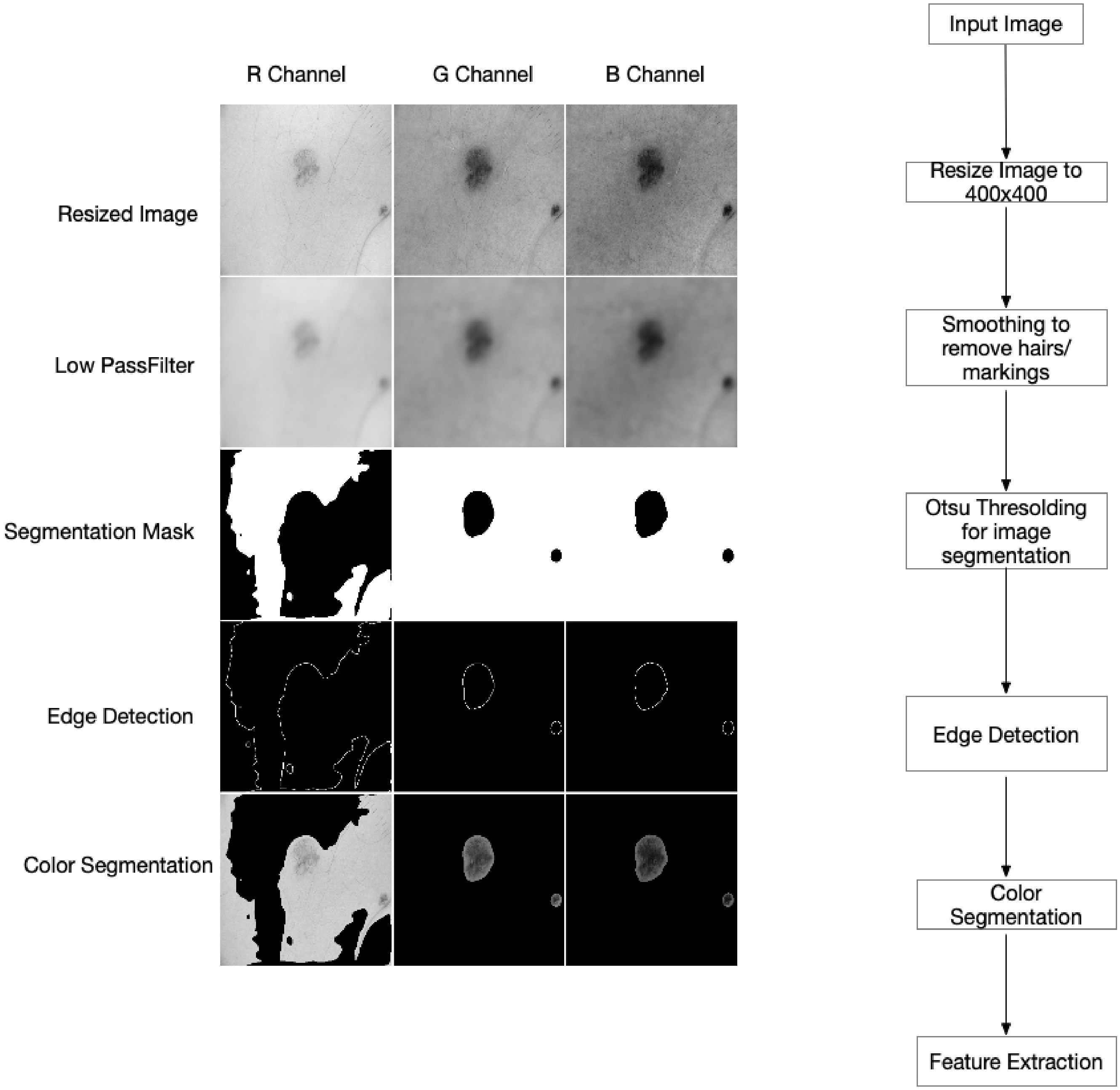


Figure 1: Feature Engineering Pipeline

Our pipeline takes an input image and prepares it for feature extraction. Once, we have our features, we then run

Table 3: RandomForest Confusion Matrix

	0	1
0	138	12
1	12	138

it through various classifiers to observe the results. All steps in the preprocessing steps are done in 3 planes. We keep results from all these planes, as the classifier should be able to choose from them. Preprocessing steps : 1. Resize all images to 400 x400 2. Convert Color to Grayscale but also preseve the color image. 3. We run it through a low pass filter to remove hairs and scale markings. 4. Similar to @SJ2015, we do automatic thresholding by Otsu in each plane which generate the mask. We apply this mask of the low pass filtered image so that we dont see any small blobs or markings inside the lesion. 5. We apply image detection on this segmented filter. 6. We take the color image and pass it through the segmented image, so we will only see color in the segmented section of the interested area.

Do notice from the image above that the red channel does not yield very good preprocessing results while the green and blue channels are very successful in isolation the regions.

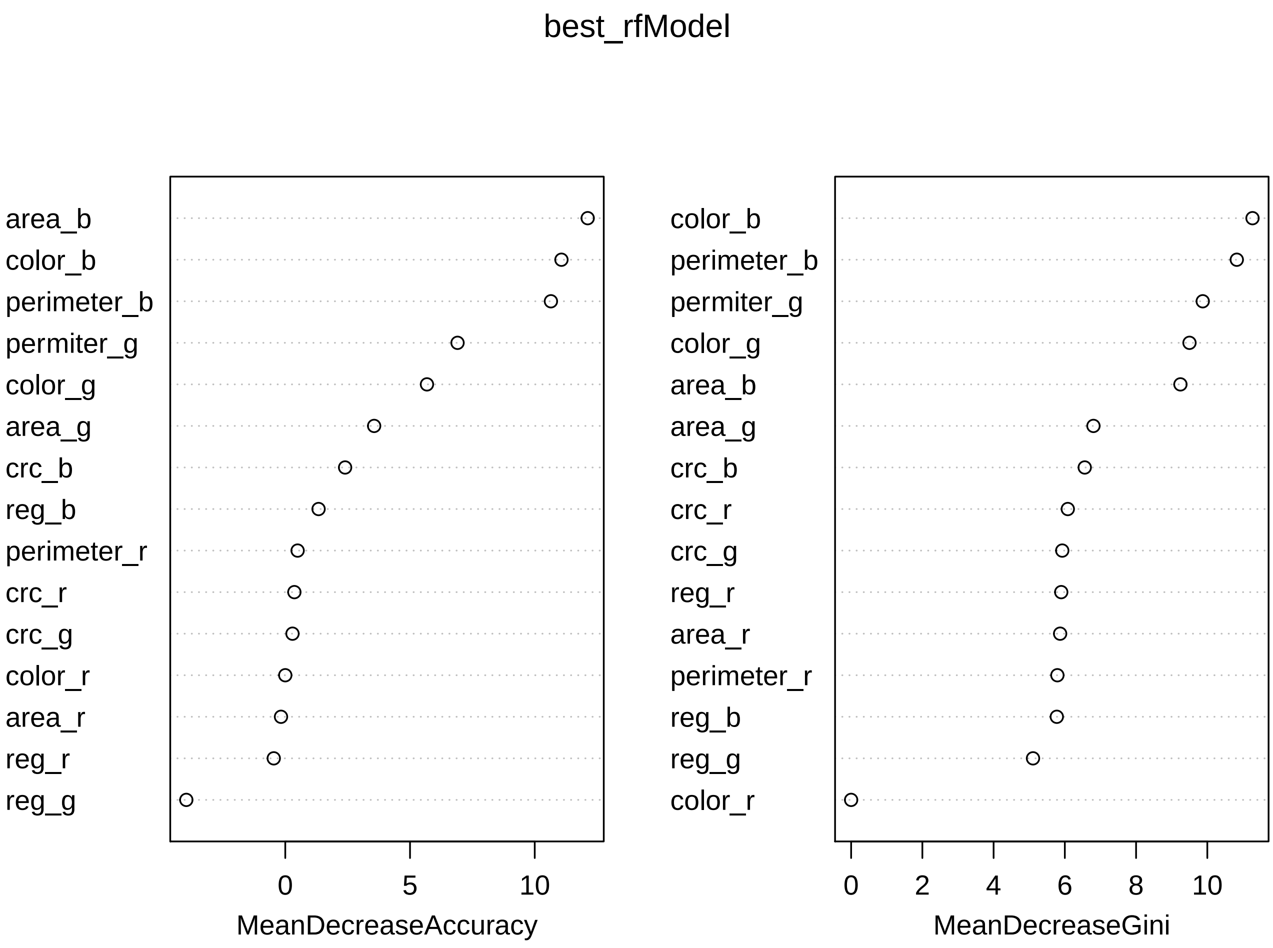
We generate 5 features for each of the channels: 1. Area: We count the number pixels in the mask of segmentation which are 0. 2. Perimeter: In the edge detected image, we count the number of pixels which are 1. 3. After applying the segmentation mask on the color image, we sum the total pixels that are in the segmented area. 4. Regularity Parameter: The ration of area / perimeter 5. Circularity Index: The ratio of (4*pi*area)/(perimeterˆ2)

[**20 Points, 2 page**] **Classification models based on new features. Fit two different classification models to identify malignant moles. You can either use the ones from Question 1 or use some new models if you believe they may perform better on the new features. Same requirements of Question 1 apply to this part. Besides, you should focus more on variable selection and interpretation.**

As with question1, we run through a random forest model with a grid search on number of tress and nodesize. The best model was with 500 trees and 5 nodesize with an accuracy of 92%.

```r
#best_rfModel <- randomForest(formula = as.factor(y) ~ ., data = train.data, importance = T, ntree=best_
yhat.all <- predict(best_rfModel, all_data_x)
temp <- cbind(as.numeric(as.character(yhat.all)), (all_data_y))
all_error <- length(which(temp[, 1] != temp[, 2])) * 100 / nrow(all_data_x)
temp <- confusionMatrix(as.factor(yhat.all), as.factor(all_data_y))
kable(temp$table, caption="RandomForest Confusion Matrix")
```

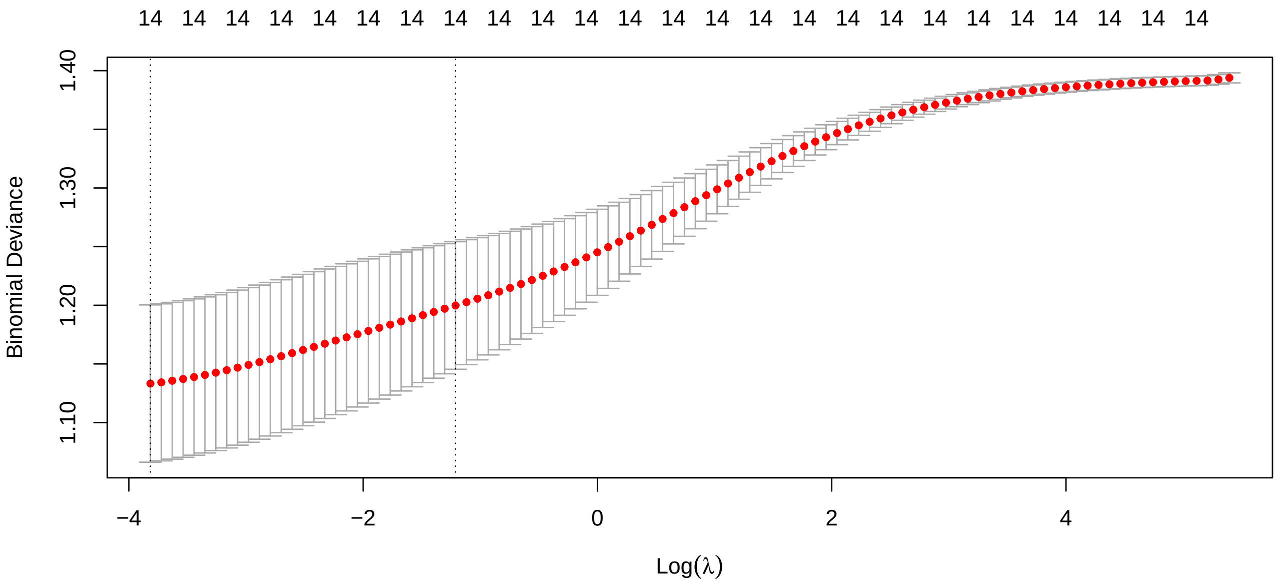
```r
varImpPlot(best_rfModel)
```



By Making understandable features, we see from above that: 1. Area, color and perimeter are key features. If the lesion contains multiple colors, then as @aad explained, it is a high chance of cancer. 2. The next set of feature are the regularity and circularity index respectively. 3. Red is the least important channel in the model. We saw during preprocessing as well, the Red plane didn not yield good results as well.

```
# use parallel for performace
  registerDoMC(cores = 4)

  # Ridge Regression
  cv_glmnet_model <- cv.glmnet(train.data[, -1], train.data[, 1], parallel = TRUE, alpha=0, family="binomi
```



```
best_lambda = cv_glmnet_model$lambda.1se
# training with best lambda selected from the cv
train_glmnet_model <- glmnet(train.data[, -1], train.data[, 1], lambda = best_lambda, alpha=0, family="b

summary(train_glmnet_model)
```

```
##            Length Class     Mode
## a0          1     -none-    numeric
## beta       15     dgCMatrix S4
## df          1     -none-    numeric
## dim         2     -none-    numeric
## lambda      1     -none-    numeric
## dev.ratio   1     -none-    numeric
## nulldev     1     -none-    numeric
## npasses     1     -none-    numeric
## jerr        1     -none-    numeric
## offset      1     -none-    logical
## classnames  2     -none-    character
## call        6     -none-    call
## nobs        1     -none-    numeric
```

```
pred <- predict(train_glmnet_model, s = best_lambda, newx = test.data[, -1], type = "class")
accuracy = mean(test.data[, 1] == pred)

results = data.frame("Best lambda" = best_lambda, "Accuracy" = accuracy)
```

```
kable(results, caption = "Ridge Regression Results")
```

Table 4: Ridge Regression Results

Best.lambda	Accuracy
0.2978745	0.7333333