edX Data-Science CYOProject

Aditya Chate

08/01/2021

Overview:

The objective of the project was to demonstrate the performance of a machine learning task on a dataset of one's own choice. The dataset selected for this purpose comprised of observational data on Indian patients of liver disease and healthy controls, obtained from the kaggle repository (https://www.kaggle.com/uciml/indian-liver-patient-records). The data was converted into .RData format and uploaded on a GitHub repository for ease of access (https://github.com/adi297/CYOProject.git).

The data has a total of 583 observations on levels of various proteins obtained from tests such as SGOT and SGPT, along with information about age and gender. The presence or absence of liver disease is coded as "1" and "2" respectively.

Based on this examination of the data, machine learning algorithms suitable for two-class classification were chosen for training. The **best reported accuracy** was **0.75** (75%). The model with the best accuracy had a **sensitivity** of **0.95** (95%) and **specificity** of **0.23** (23%).

An **ensemble** of the models used generated an **accuracy** of **0.74** (74%). The **overall mean accuracy** was **0.73** (73%) with a **standard deviation** of **0.015**.

Methodology and Analysis:

Structure of the dataset:

```
## Classes ’data.table’ and ’data.frame’:   583 obs. of  11 variables:
##  $ Age                       : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                    : chr  "Female" "Male" "Male" "Male" ...
##  $ Total_Bilirubin           : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ Direct_Bilirubin          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ Alkaline_Phosphotase      : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ Alamine_Aminotransferase  : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ Aspartate_Aminotransferase: int  18 100 68 20 59 14 12 11 19 58 ...
##  $ Total_Protiens            : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ Albumin                   : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ Albumin_and_Globulin_Ratio: num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Dataset                   : int  1 1 1 1 1 1 1 1 2 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

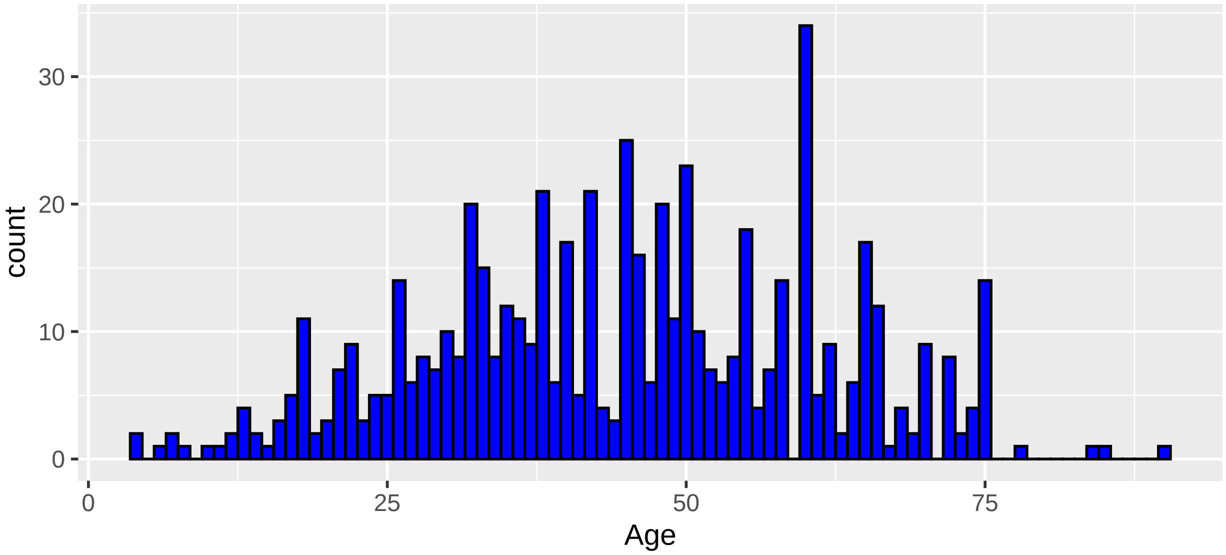
The dataset shows observations on blood proteins and age as numeric variables, gender as a character variable, and the presence or absence of liver disease coded as “1” and “2” respectively under a numeric variable called *Dataset*.

First six rows of the dataset:

```
##    Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase
## 1:  65 Female             0.7              0.1                  187
## 2:  62   Male            10.9              5.5                  699
## 3:  62   Male             7.3              4.1                  490
## 4:  58   Male             1.0              0.4                  182
## 5:  72   Male             3.9              2.0                  195
## 6:  46   Male             1.8              0.7                  208
##    Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
## 1:                       16                         18            6.8     3.3
## 2:                       64                        100            7.5     3.2
## 3:                       60                         68            7.0     3.3
## 4:                       14                         20            6.8     3.4
## 5:                       27                         59            7.3     2.4
## 6:                       19                         14            7.6     4.4
##    Albumin_and_Globulin_Ratio Dataset
## 1:                       0.90       1
## 2:                       0.74       1
## 3:                       0.89       1
## 4:                       1.00       1
## 5:                       0.40       1
## 6:                       1.30       1
```

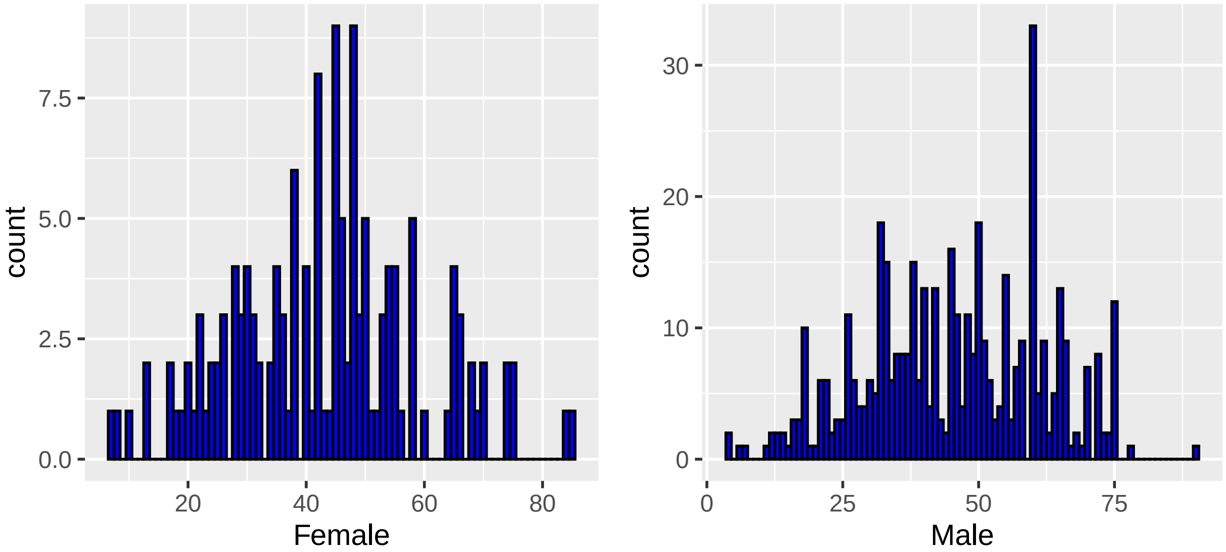
Based on a preliminary examination of the dataset, one could consider examining effects of age and gender on the prevalence of liver disease.

Age distribution observed in the dataset:



The *Age* variable, by itself, seems to show some significant spikes in distribution, yet no definite pattern that could be used to model effects in liver disease predictions.

Age distribution based on gender:



The *Age* variable, when grouped based on gender, seems to have a quasi-normal distribution for females, but no distinct pattern for males. This might just be a chance occurrence considering the small size of the dataset and differences in the prevalence of each gender.

Thus, it was decided to determine prevalence based on gender and quantify it as the odds ratio for each gender, thus resulting in a numeric quantity which would be further used with machine learning models.

Gender	Count	Diseased	Healthy	Odds_Ratio
Female	142	92	50	1.840000
Male	441	324	117	2.769231

Creating a dataset suitable for machine learning algorithms:

The available data on various protein levels, along with a numeric vector having the odds ratio as a quantification of gender effect, and the presence or absence of liver disease coded as “1” and “2” in factor form was used to generate a dataset ready for machine learning algorithm training. The latter was stored as a factor vector, while the rest of the information was put in numeric matrix with the appropriate data wrangling to generate a dataset with the following structure:

Structure of the dataset to be used for training machine learning algorithms:

```r
str(liv_exp_set)
```

```
## List of 2
##  $ x: num [1:583, 1:9] 65 62 62 58 72 46 26 29 17 55 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:9] "Age" "Total_Bilirubin" "Direct_Bilirubin" "Alkaline_Phosphotase" ...
##  $ y: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 2 1 ...
```

Calculations of distance between the binary outcomes:

The mean distance between predictors for cases *without liver disease* was **2.94**.

```r
mean(dist_2to2[2:length(dist_2to2)])
```

```
## [1] 2.939958
```

The mean distance between predictors for cases *without liver disease* and *with liver disease* was **3.62**.

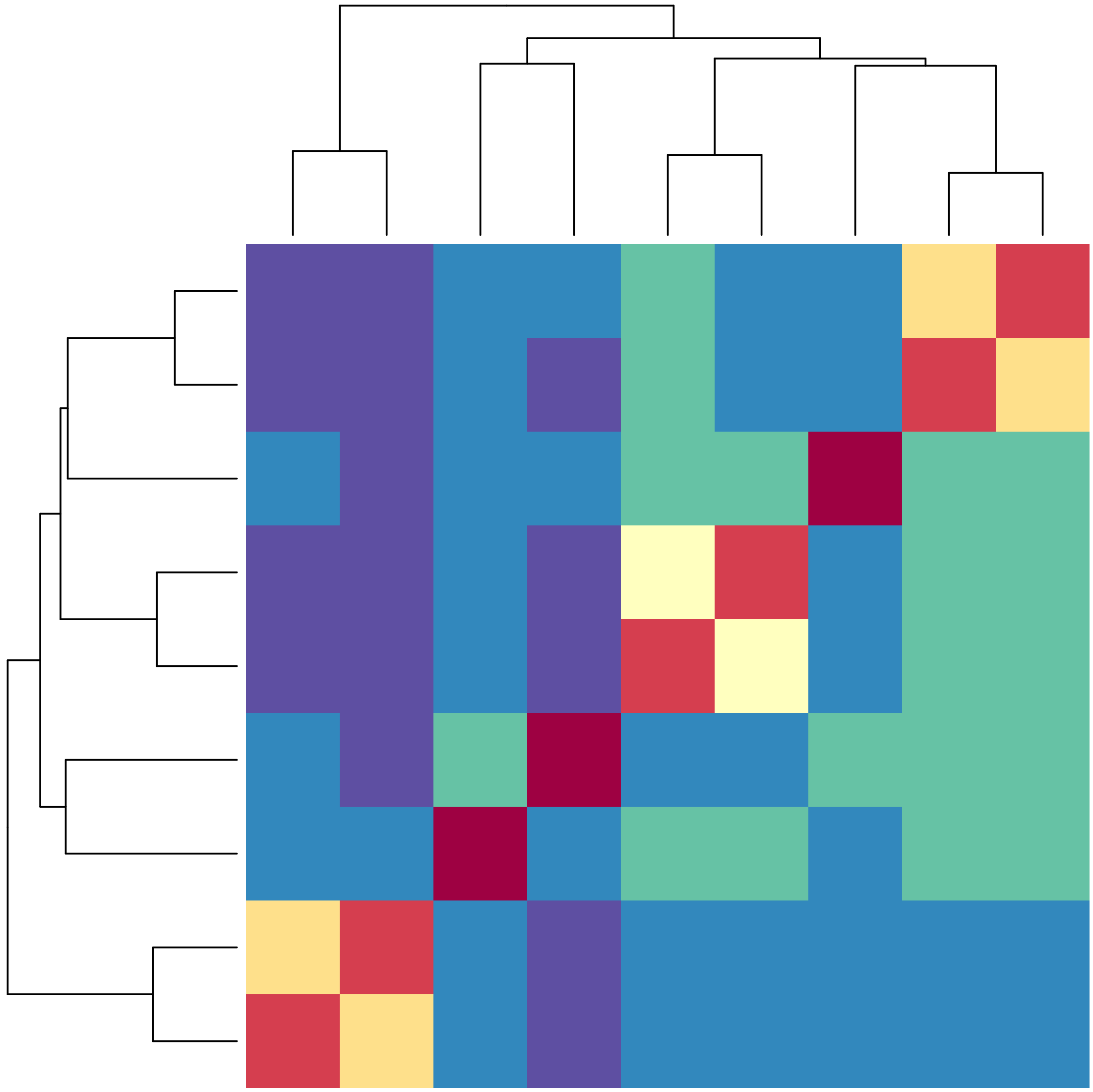
```r
mean(dist_2to1)
```

```
## [1] 3.624778
```

The rather small difference in the distance between the predictors for either cases is potential indicator of the accuracy of prediction models being sub-optimal.

Scaling the predictors revealed the following distribution for distance as can be seen in the following heatmap:

Heatmap of distances between scaled predictors:



Principal Component Analysis:

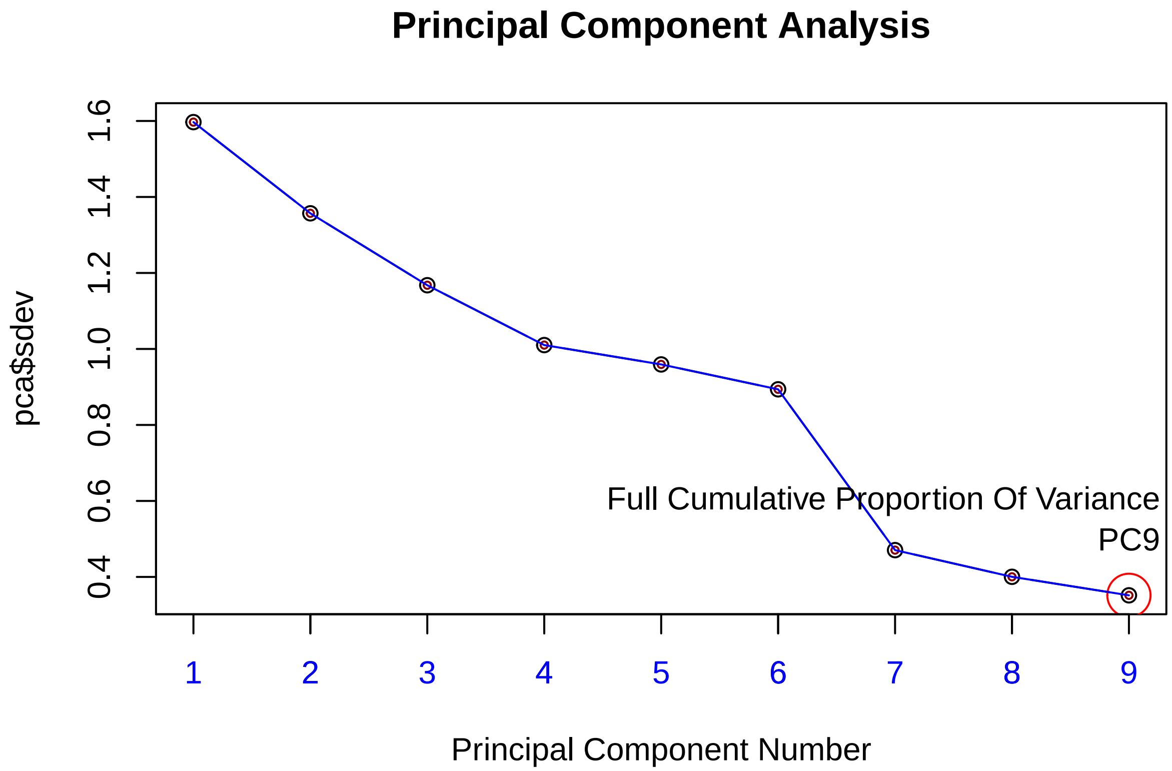
Principal Component Analysis of the scaled matrix of predictors revealed nine principal components to account for all variability in the data.

PCA Summary:

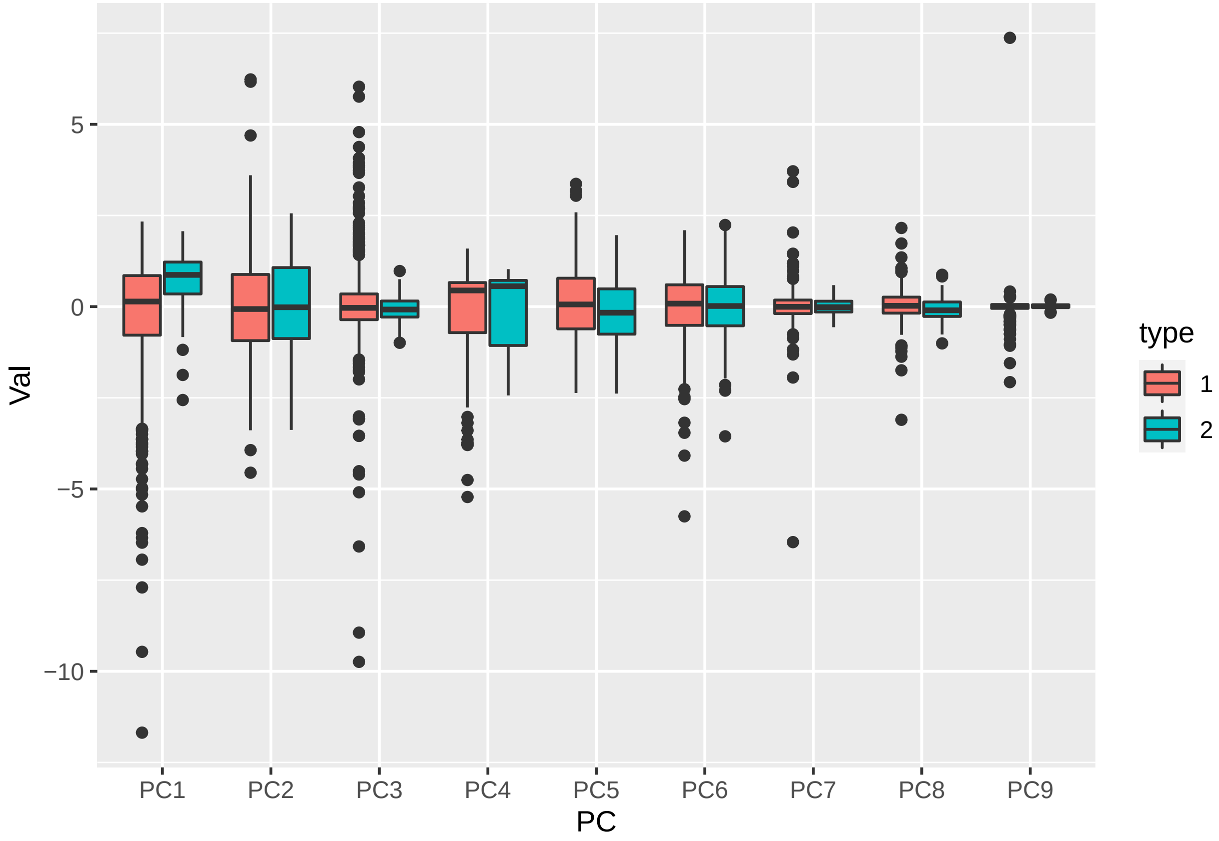
```r
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6    PC7
## Standard deviation     1.5970 1.3570 1.1677 1.0102 0.9593 0.89377 0.4705
## Proportion of Variance 0.2834 0.2046 0.1515 0.1134 0.1022 0.08876 0.0246
## Cumulative Proportion  0.2834 0.4880 0.6395 0.7529 0.8551 0.94385 0.9685
##                            PC8     PC9
## Standard deviation     0.40037 0.35163
## Proportion of Variance 0.01781 0.01374
## Cumulative Proportion  0.98626 1.00000
```

PCA Plot:



PCA Boxplot grouped by presence or absence of disease:



The data was then divided into a training and test set with an 80/20 split which ensured equal proportions of cases without liver disease and cases with liver disease in them, thus making it suitable for algorithm training.

```
#Training and tests sets both contain approximately equal proportions of
#cases with and without liver disease:

mean(train_set$y == 2)
```

```
## [1] 0.2860215
```

```
mean(test_set$y == 2)
```

```
## [1] 0.2881356
```

Machine Learning Algorithm Training:

With the objective of *binary classification* (presence or absence of liver disease), machine learning algorithms suitable for the purpose were chosen.

The first was a k-means algorithm made with a user-defined function:

```r
#Predictions based on k-means:
predict_kmeans <- function(x, k) {
  centers <- k$centers
  distances <- sapply(1:nrow(x), function(i){
    apply(centers, 1, function(y) dist(rbind(x[i,], y)))
  })
  max.col(-t(distances))
}

set.seed(1, sample.kind = "Rounding")
k <- kmeans(train_x, centers = 13)

res_kmeans <- predict_kmeans(test_x, k)
pred_kmeans <- ifelse(res_kmeans == 1, "2", "1")
pred_kmeans <- as.factor(pred_kmeans)
cm_kmeans <- confusionMatrix(pred_kmeans, test_y)
cm_kmeans$overall["Accuracy"]
```

The following models were used without any optimization of tuning parameters:

- Boosted Classification Trees (ada)
- AdaBoost Classification Trees (adaboost)
- Distance Weighted Discrimination with Radial Basis Function Kernel (dwdRadial)

The following models were used with optimization of tuning parameters:

- k - nearest neighbors (knn)
- RandomForest (rf)
- Oblique Random Forest (ORFpls)

Tuning Parameter Optimization:

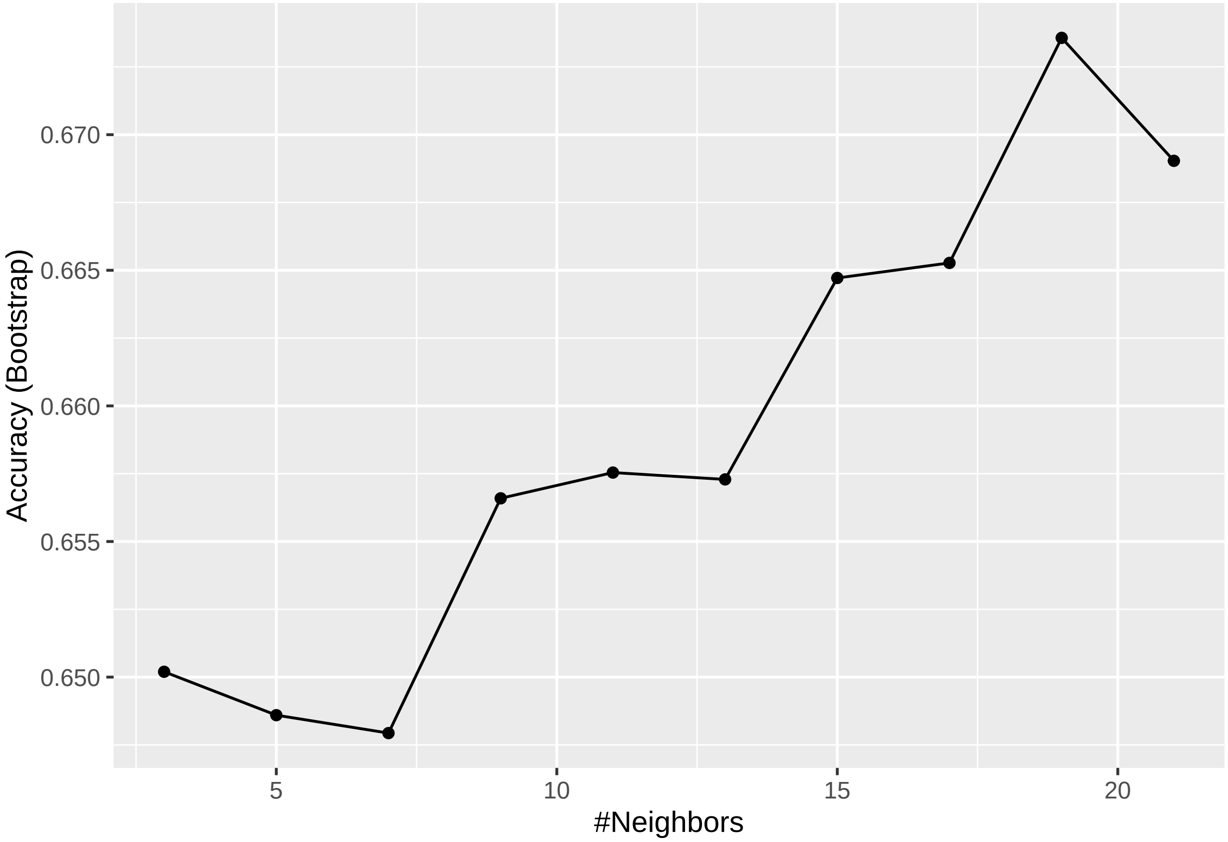
Optimization of tuning parameters is demonstrated in the following graphs:

knn Optimization:

```
#Optimal k:
best_k <- knn_fit$bestTune
best_k
```

```
##    k
## 9 19
```

```
#k Optimization Plot:
ggplot(knn_fit)
```



randomForest Optimization:

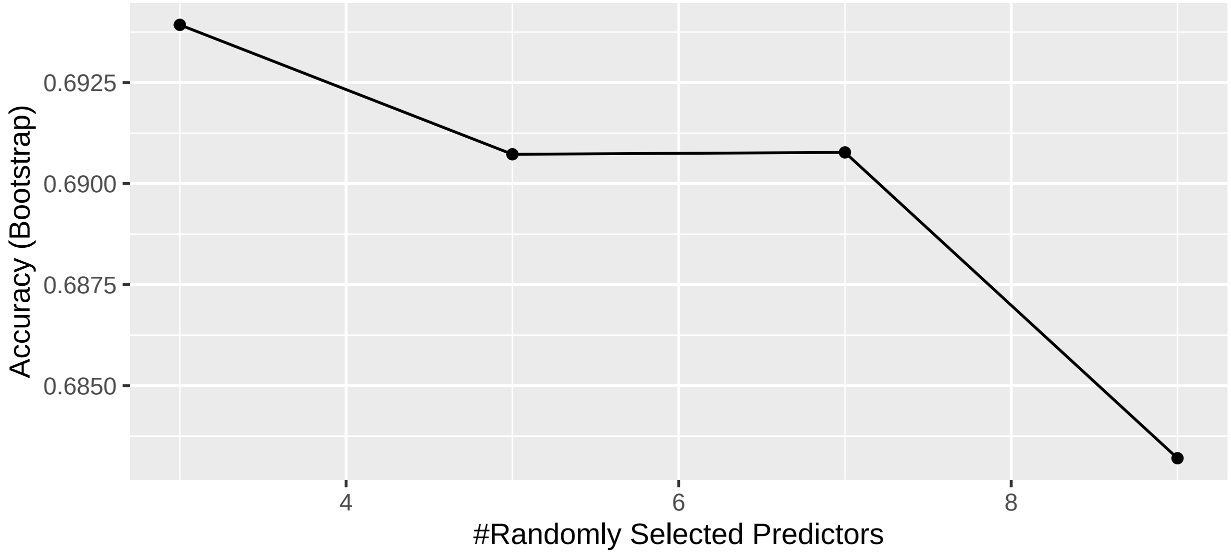
```
#Optimal predictors:
best_mtry_rf <- rf_fit$bestTune
best_mtry_rf
```

```
##   mtry
## 1    3
```

```
#Variable Importance:
varImp(rf_fit)
```

```
## rf variable importance
##
##                               Importance
## x.Direct_Bilirubin               100.000
## x.Total_Bilirubin                 72.504
## x.Alamine_Aminotransferase        62.155
## x.Age                             57.023
## x.Aspartate_Aminotransferase       54.193
## x.Alkaline_Phosphotase            51.788
## x.Albumin                         20.247
## x.Odds_Ratio                       1.975
## x.Total_Protiens                   0.000
```

```
#Predictor number optimization:
ggplot(rf_fit)
```



Oblique randomForest Optimization:

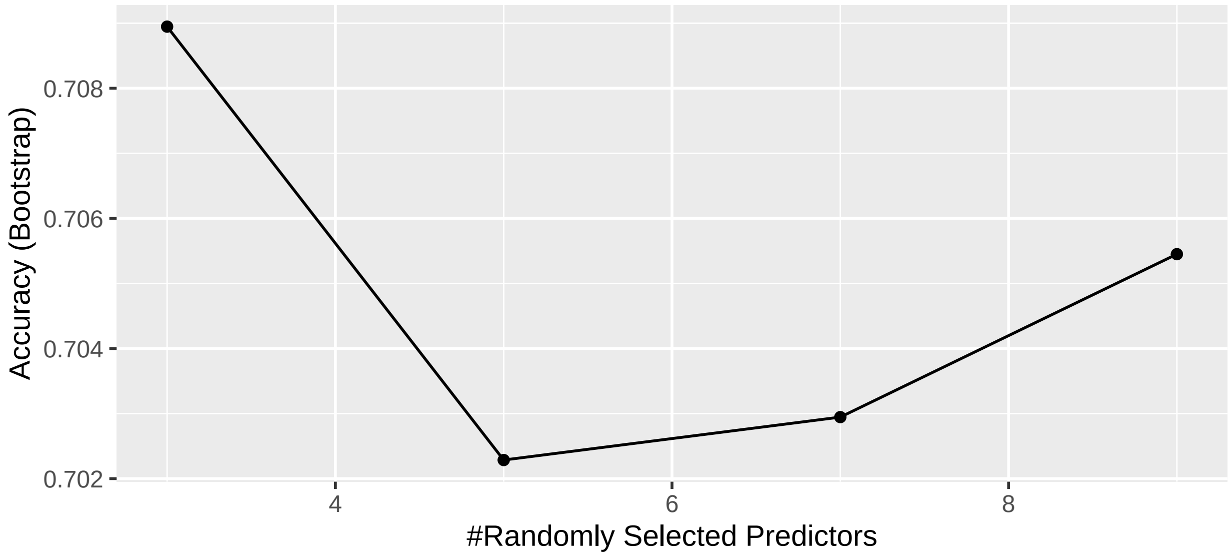
```
#Optimal predictors:
best_mtry_ORFpls <- ORFpls_fit$bestTune
best_mtry_ORFpls
```

```
##   mtry
## 1    3
```

```
#Variable Importance:
varImp(ORFpls_fit)
```

```
## ROC curve variable importance
##
##                              Importance
## x.Total_Bilirubin               100.000
## x.Aspartate_Aminotransferase     97.605
## x.Direct_Bilirubin               94.795
## x.Alkaline_Phosphotase           90.509
## x.Alamine_Aminotransferase       90.174
## x.Albumin                        55.477
## x.Age                            49.856
## x.Odds_Ratio                      6.681
## x.Total_Protiens                  0.000
```

```
#Predictor number optimization:
ggplot(ORFpls_fit)
```



Results:

The **best accuracy** was obtained from the **"adaboost"** model: **0.75**
The **ensemble** yielded an accuracy of: **0.74**

The **overall mean accuracy** was: **0.73**
with a **standard deviation** of: **0.015**

Final Results:

```
#Final result table:
results %>% knitr::kable()
```

models	accuracies
kmeans	0.73
ada	0.75
adaboost	0.75
dwdRadial	0.71
knn	0.72
rf	0.72
ORFpls	0.72
ensemble	0.74

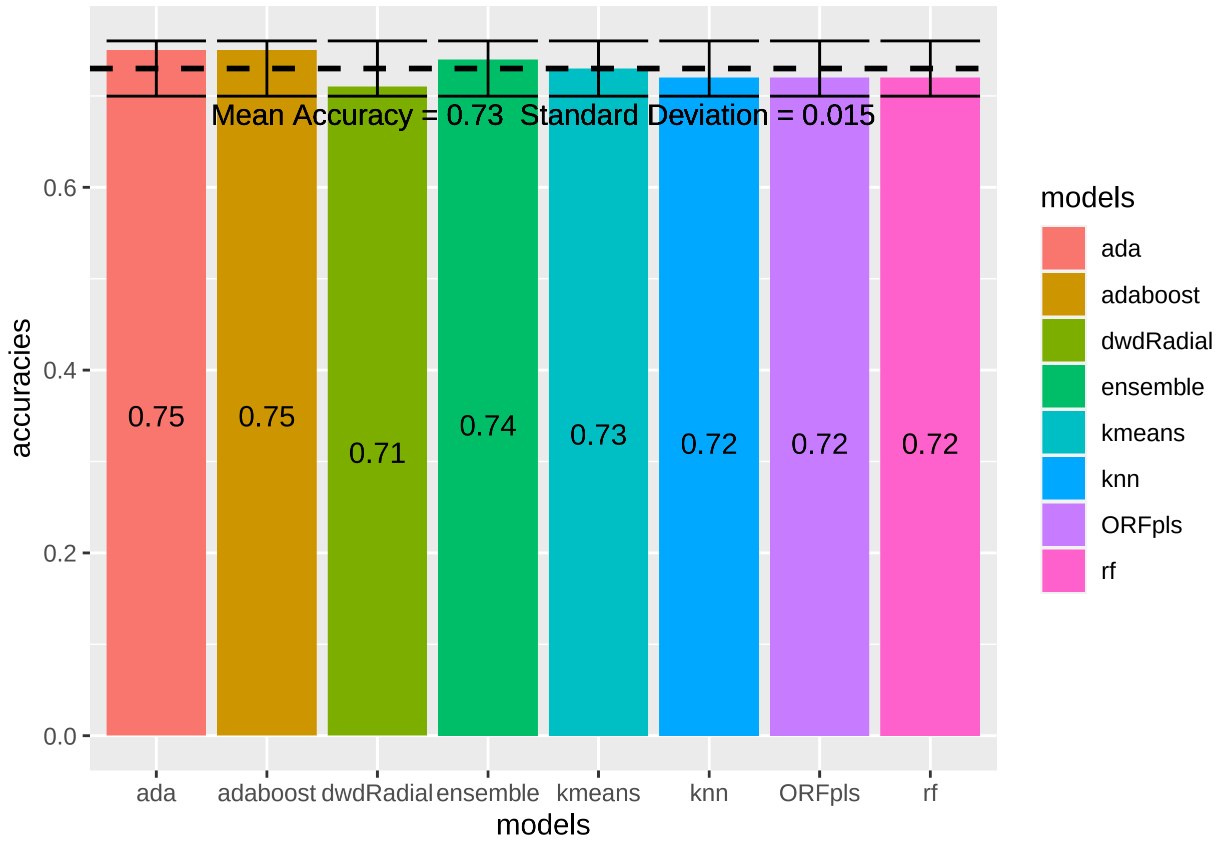
```
#Overall mean accuracy of all models:
final_acc <- mean(results$accuracies)
final_acc
```

```
## [1] 0.73
```

```
#Overall standard deviation of all models:
sd_acc <- sd(results$accuracies)
sd_acc
```

```
## [1] 0.01511858
```

```r
#Results Graph:
results_graph <- results %>% ggplot(aes(models, accuracies, fill = models)) +
            geom_col() +
            geom_hline(yintercept = 0.73, lty = 2, size = 1) +
            geom_text(aes(x = 0, y = 0.66,
                          label = "Mean Accuracy = 0.73  Standard Deviation = 0.015"),
                      nudge_x = 4.5, nudge_y = 0.02) +
            geom_text(aes(label = accuracies), nudge_y = -0.4) +
            geom_errorbar(aes(ymin = final_acc - 2*sd_acc,
                              ymax = final_acc + 2*sd_acc))
results_graph
```



Best model (AdaBoost Classification Trees) confusionMatrix:

- Accuracy: 0.75
- Sensitivity: 0.95
- Specificity: 0.23

```
cm_adaboost
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 80 26
##          2  4  8
##
##                Accuracy : 0.7458
##                  95% CI : (0.6574, 0.8214)
##     No Information Rate : 0.7119
##     P-Value [Acc > NIR] : 0.240701
##
##                   Kappa : 0.2324
##
##  Mcnemar's Test P-Value : 0.000126
##
##             Sensitivity : 0.9524
##             Specificity : 0.2353
##          Pos Pred Value : 0.7547
##          Neg Pred Value : 0.6667
##              Prevalence : 0.7119
##          Detection Rate : 0.6780
##    Detection Prevalence : 0.8983
##       Balanced Accuracy : 0.5938
##
##        'Positive' Class : 1
##
```

Conclusion:

Machine learning algorithms typically use large datasets to have better accuracy of prediction by means of having sufficient data to train on. However, when it comes to patient records, confidentiality is an issue. This might limit the access that is available to patient information. Published results of large clinical trials or observational studies often end up including large quantities of *NA*s in their data due to irregularity in the behavior of patients/volunteers for a study.

Thus, when limited to small sample sizes, accuracy and other metrics of machine learning algorithms may not be optimal. The rather small difference (0.32) in the distance between the predictors for cases without liver disease, and the distance between predictors for cases without liver disease and with liver disease, potentially indicated sub-optimal efficiency of machine learning models. Yet, an **overall mean accuracy** of **73%** with a **standard deviation** as low as **0.015**, and the **best model** showing an **accuracy** of **75%** with **95% sensitivity** and **23% specificity** could be considered as a promising algorithm for scaling up further with more data.

Future work could involve scaling up to larger data sizes and further optimization of tuning parameters to obtain more accuracy of predictability, and better sensitivity and specificity.