Lab 8

Nur Izzati Binti Shalahudin

2022-05-13

Get dataset & clean it

```
data("BreastCancer")
breast <- BreastCancer

#remove 1st column (ID column, not needed)
breast <- select(breast, -1)

#impute NA values in bare.nuclei into mean value
breast$Bare.nuclei<-as.numeric(breast$Bare.nuclei)
breast$Bare.nuclei[is.na(breast$Bare.nuclei)] <-round(mean(breast$Bare.nuclei, na.rm = T),digits = 0)

#change 'Class' into factors
breast$Class<-as.character(breast$Class)
breast$Class<-as.factor(breast$Class)

# set seed
set.seed(7)

# traincontrol
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "Accuracy"
```

Linear Algorithm

Linear Regression (LR)

```
fit.glm= train(Class~.,
               data=breast,
               method="glm",
               metric=metric,
               trControl=trainControl)
```

Linear Discriminate Analysis (LDA)

```
lda.fit = train(Class ~ ., data=breast, method="lda",metric=metric,
                trControl=trainControl)
```

Regularized Logistic Regression (GLMNET)

```
glmnet.fit<-train(Class ~ ., data=breast, method="glmnet",metric=metric,
           trControl=trainControl)
```

Non-Linear Algorithm

k-Nearest Neighbors (kNN)

```
knn.fit<-train(Class ~ ., data=breast, method="knn",metric=metric,
                  trControl=trainControl)
```

Classification and Regression Trees (CART)

```
cart.fit<-train(Class ~ ., data=breast, method="rpart",metric=metric,
               trControl=trainControl)
```

Naive Bayes (NB)

```
nb.fit<-train(Class ~ ., data=breast, method="naive_bayes",metric=metric,
                trControl=trainControl)
```

Result

```
print(fit.glm)
```

```
## Generalized Linear Model
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 629, 629, 629, 629, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9174937  0.8139778
```

```
print(lda.fit)
```

```
## Linear Discriminant Analysis
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 629, 629, 629, 629, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9618283  0.9149981
```

```
print(glmnet.fit)
```

```
## glmnet
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 629, 629, 629, 630, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        Accuracy   Kappa
##   0.10   0.0007784717  0.9523234  0.8935106
##   0.10   0.0077847173  0.9628207  0.9175168
##   0.10   0.0778471727  0.9647049  0.9222102
##   0.55   0.0007784717  0.9532758  0.8955857
##   0.55   0.0077847173  0.9623240  0.9164982
##   0.55   0.0778471727  0.9566091  0.9032676
##   1.00   0.0007784717  0.9528063  0.8941919
##   1.00   0.0077847173  0.9642422  0.9203591
##   1.00   0.0778471727  0.9523232  0.8921014
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 0.1 and lambda = 0.07784717.
```

```
print(knn.fit)
```

```
## k-Nearest Neighbors
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
```

```
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 629, 629, 629, 629, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9365890  0.8560081
##   7  0.9375343  0.8576574
##   9  0.9384800  0.8598246
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
print(cart.fit)
```

```
## CART
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 630, 629, 629, 629, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.02489627  0.9375412  0.8631231
##   0.05394191  0.9260712  0.8400167
##   0.78008299  0.8280472  0.5503355
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.02489627.
```

```
print(nb.fit)
```

```
## Naive Bayes
##
## 699 samples
##   9 predictor
##   2 classes: 'benign', 'malignant'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 629, 629, 630, 629, 629, 629, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9666373  0.9274946
##    TRUE      0.9537798  0.8970641
##
## Tuning parameter 'laplace' 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 laplace = 0, usekernel = FALSE
##  and adjust = 1.
```

Ways to improve performance of model

1.Add more data for malignant cases (bcs benign cases have 2x more data)

2.Ensure the data collected are complete (no NA/NAN)