CensusPay.R

rharidas

2021-11-19

```
# Execute the given source code for the project
source("DatasetProcessingCode.R")
```

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used

## Rows: 32,561
## Columns: 15
## $ age            <int> 90, 82, 66, 54, 41, 34, 38, 74, 68, 41, 45, 38, 52, 32,~
## $ workclass      <chr> "?", "Private", "?", "Private", "Private", "Private", "~
## $ fnlwgt         <int> 77053, 132870, 186061, 140359, 264663, 216864, 150601, ~
## $ education      <chr> "HS-grad", "HS-grad", "Some-college", "7th-8th", "Some-~
## $ education.num  <int> 9, 9, 10, 4, 10, 9, 6, 16, 9, 10, 16, 15, 13, 14, 16, 1~
## $ marital.status <chr> "Widowed", "Widowed", "Widowed", "Divorced", "Separated~
## $ occupation     <chr> "?", "Exec-managerial", "?", "Machine-op-inspct", "Prof~
## $ relationship   <chr> "Not-in-family", "Not-in-family", "Unmarried", "Unmarri~
## $ race           <chr> "White", "White", "Black", "White", "White", "White", "~
## $ sex            <chr> "Female", "Female", "Female", "Female", "Female", "Fema~
## $ capital.gain   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ capital.loss   <int> 4356, 4356, 4356, 3900, 3900, 3770, 3770, 3683, 3683, 3~
## $ hours.per.week <int> 40, 18, 40, 40, 40, 45, 40, 20, 40, 60, 35, 45, 20, 55,~
## $ native.country <chr> "United-States", "United-States", "United-States", "Uni~
## $ income         <chr> "<=50K", "<=50K", "<=50K", "<=50K", "<=50K", "<=50K", "~
## Rows: 29,170
## Columns: 13
## $ age           <int> 90, 82, 66, 54, 41, 34, 38, 74, 68, 45, 38, 52, 32, 51, ~
## $ fnlwgt        <int> 77053, 132870, 186061, 140359, 264663, 216864, 150601, 8~
## $ education     <fct> HSgrad, HSgrad, Somecollege, 7th8th, Somecollege, HSgrad~
## $ eduyears      <int> 9, 9, 10, 4, 10, 9, 6, 16, 9, 16, 15, 13, 14, 16, 15, 7,~
## $ maritalstatus <fct> Widowed, Widowed, Widowed, Divorced, Separated, Divorced~
## $ occupation    <fct> Unknown, Execmanagerial, Unknown, Machineopinspct, Profs~
## $ relationship  <fct> Notinfamily, Notinfamily, Unmarried, Unmarried, Ownchild~
## $ race          <fct> White, White, Black, White, White, White, White, White, ~
## $ sex           <fct> Female, Female, Female, Female, Female, Female, Male, Fe~
## $ hoursperweek  <int> 40, 18, 40, 40, 40, 45, 40, 20, 40, 35, 45, 20, 55, 40, ~
## $ native        <chr> "UnitedStates", "UnitedStates", "UnitedStates", "UnitedS~
## $ income        <fct> AtBelow50K, AtBelow50K, AtBelow50K, AtBelow50K, AtBelow5~
## $ class         <fct> Unknown, Private, Unknown, Private, Private, Private, Pr~
```

```
library(caret)
library(gridExtra)
library(kableExtra)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(1996,sample.kind="Rounding")
```

```
## Warning in set.seed(1996, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
#the simplest possible machine algorithm: guessing the outcome
seat_of_the_pants <- sample(c("Above50K", "AtBelow50K"), length(test_index), replace = TRUE) %>% factor
accuracy_guess <- mean(seat_of_the_pants == adultpayclean_validation$income)

#build a confusion matrix for this simple model
table(predicted = seat_of_the_pants, actual = adultpayclean_validation$income)
```

```
##             actual
## predicted    Above50K AtBelow50K
##   Above50K        347       1087
##   AtBelow50K      371       1113
```

```
#tabulate accuracy by income levels
adultpayclean_validation %>%
  mutate(y_hat = seat_of_the_pants) %>%
  group_by(income) %>%
  summarize(accuracy = mean(y_hat == income))
```

```
## # A tibble: 2 x 2
##   income     accuracy
##   <fct>         <dbl>
## 1 Above50K      0.483
## 2 AtBelow50K    0.506
```

```r
# confusion matrix using R function
cm <- confusionMatrix(data =seat_of_the_pants , reference = adultpayclean_validation$income)
cm
```

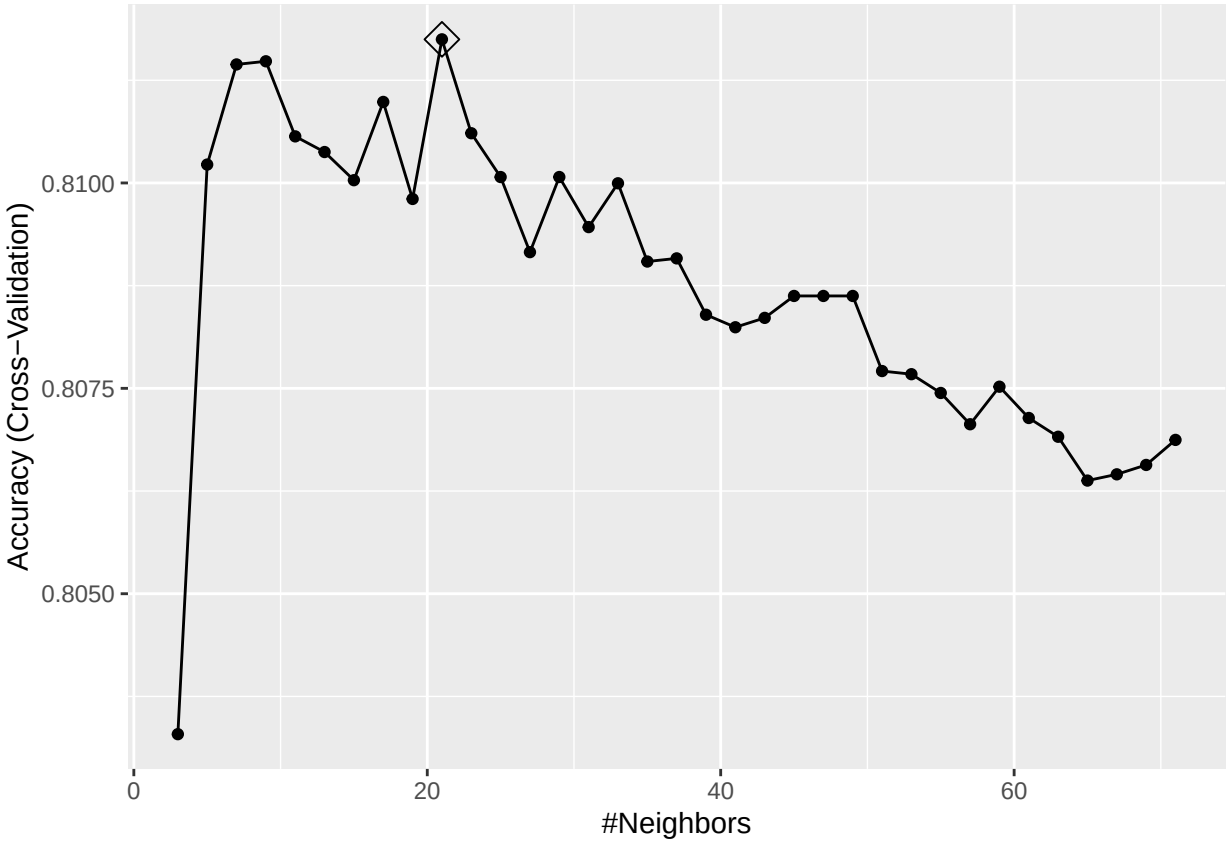
```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Above50K AtBelow50K
##   Above50K        347       1087
##   AtBelow50K      371       1113
##
##                Accuracy : 0.5003
##                  95% CI : (0.482, 0.5186)
##     No Information Rate : 0.7539
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.0081
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4833
##             Specificity : 0.5059
##          Pos Pred Value : 0.2420
##          Neg Pred Value : 0.7500
##              Prevalence : 0.2461
##          Detection Rate : 0.1189
##    Detection Prevalence : 0.4914
##       Balanced Accuracy : 0.4946
##
##        'Positive' Class : Above50K
##
```

```r
sensitivity_guess <- cm$byClass[["Sensitivity"]]
specificity_guess <- cm$byClass[["Specificity"]]
prevalence_guess <- cm$byClass[["Prevalence"]]

#logistic linear model
# create the model
lm_fit <- adultpayclean_train %>%
    mutate(y = as.numeric(income == "Above50K")) %>%
    lm(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, data=.)

# predict using test set
p_hat_logit <- predict(lm_fit, newdata = adultpayclean_validation)

#translate predicted data into factor
y_hat_logit <- ifelse(p_hat_logit > 0.5, "Above50K", "AtBelow50K") %>% factor

#compare the predicted vs observed values and use confusionMatrix to get the accuracy and other metrics
cm_lm <- confusionMatrix(y_hat_logit, adultpayclean_validation$income)
accuracy_lm <- confusionMatrix(y_hat_logit, adultpayclean_validation$income)$overall[["Accuracy"]]

sensitivity_lm <- cm_lm$byClass[["Sensitivity"]]
```

```r
specificity_lm <- cm_lm$byClass[["Specificity"]]
prevalence_lm <- cm_lm$byClass[["Prevalence"]]

#general linear model
#create the glm model
glm_fit <- adultpayclean_train %>%
  mutate(y = as.numeric(income == "Above50K")) %>%
  glm(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, data=., family = "

# predict using validation set
p_hat_logit <- predict(glm_fit, newdata = adultpayclean_validation)

# translate the predicted data into factor
y_hat_logit <- ifelse(p_hat_logit > 0.5, "Above50K", "AtBelow50K") %>% factor

# compare the predicted vs observed values and use confusionMatrix to get the accuracy and other metric
cm_glm <- confusionMatrix(y_hat_logit, adultpayclean_validation$income)
accuracy_glm <- confusionMatrix(y_hat_logit, adultpayclean_validation$income)$overall[["Accuracy"]]

sensitivity_glm <- cm_glm$byClass[["Sensitivity"]]
specificity_glm <- cm_glm$byClass[["Specificity"]]
prevalence_glm <- cm_glm$byClass[["Prevalence"]]

# translate income factor into binary outcome
temp <- adultpayclean_train %>%
  mutate(y = as.factor(income == "Above50K"))

#k-nearest neighbors with a train control and tuning
set.seed(2008)
# train control to use 10% of the observations each to speed up computations
control <- trainControl(method = "cv", number = 10, p = .9)
# train the model using knn. choose the best k value using tuning algorithm
train_knn <-  train(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, metho
                    data = temp,
                    tuneGrid = data.frame(k = seq(3, 71, 2)),trControl = control)
```

```
## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```

```r
#plot the resulting model
ggplot(train_knn, highlight = TRUE)
```




```
#verify which k value was used
train_knn$bestTune
```

```
##    k
## 10 21
```

```
train_knn$finalModel
```

```
## 21-nearest neighbor model
## Training set outcome distribution:
##
## FALSE  TRUE
## 19799  6453
```

```
#use this trained model to predict raw knn predictions
y_hat_knn <- predict(train_knn,adultpayclean_validation, type = "raw")

# compare the predicted and observed values using confusionMatrix to get the accuracy and other metrics
cm_knn <- confusionMatrix(y_hat_knn, as.factor(adultpayclean_validation$income == "Above50K"))
accuracy_knn <- confusionMatrix(y_hat_knn, as.factor(adultpayclean_validation$income == "Above50K"))$ove

sensitivity_knn <- cm_knn$byClass[["Sensitivity"]]
specificity_knn <- cm_knn$byClass[["Specificity"]]
prevalence_knn <- cm_knn$byClass[["Prevalence"]]
```

```r
#k-nearest classification using tuning function
set.seed(2008)
ks <- seq(3, 251, 2)
knntune <- map_df(ks, function(k){

  temp <- adultpayclean_train %>%
    mutate(y = as.factor(income == "Above50K"))
  temp_test <- adultpayclean_validation %>%
    mutate(y = as.factor(income == "Above50K"))

  knn_fit <- knn3(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, data =

  y_hat <- predict(knn_fit, temp, type = "class")
  cm_train <- confusionMatrix(y_hat, temp$y)
  train_error <- cm_train$overall["Accuracy"]

  y_hat <- predict(knn_fit, temp_test, type = "class")
  cm_test <- confusionMatrix(y_hat, temp_test$y)
  test_error <- cm_test$overall["Accuracy"]

  tibble(train = train_error, test = test_error)
})

accuracy_knntune <- max(knntune$test)

knn_fit <- knn3(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, data = te

y_hat <- predict(knn_fit, temp, type = "class")
cm_knntune <- confusionMatrix(y_hat, temp$y)

sensitivity_knntune <- cm_knntune$byClass[["Sensitivity"]]
specificity_knntune <- cm_knntune$byClass[["Specificity"]]
prevalence_knntune <- cm_knntune$byClass[["Prevalence"]]

#k-nearest using knn3
set.seed(2008)
knn3_fit <- knn3(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, data = t
y_hat_knn3 <- predict(knn3_fit, adultpayclean_validation, type = "class")

cm_knn3 <- confusionMatrix(y_hat_knn3, as.factor(adultpayclean_validation$income  == "Above50K"))
accuracy_knn3 <- confusionMatrix(y_hat_knn3, as.factor(adultpayclean_validation$income  == "Above50K"))$

sensitivity_knn3 <- cm_knn3$byClass[["Sensitivity"]]
specificity_knn3 <- cm_knn3$byClass[["Specificity"]]
prevalence_knn3 <- cm_knn3$byClass[["Prevalence"]]


#recursive partitioning using rpart
set.seed(2008)
train_rpart <- train(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)),
                     data = temp)
```

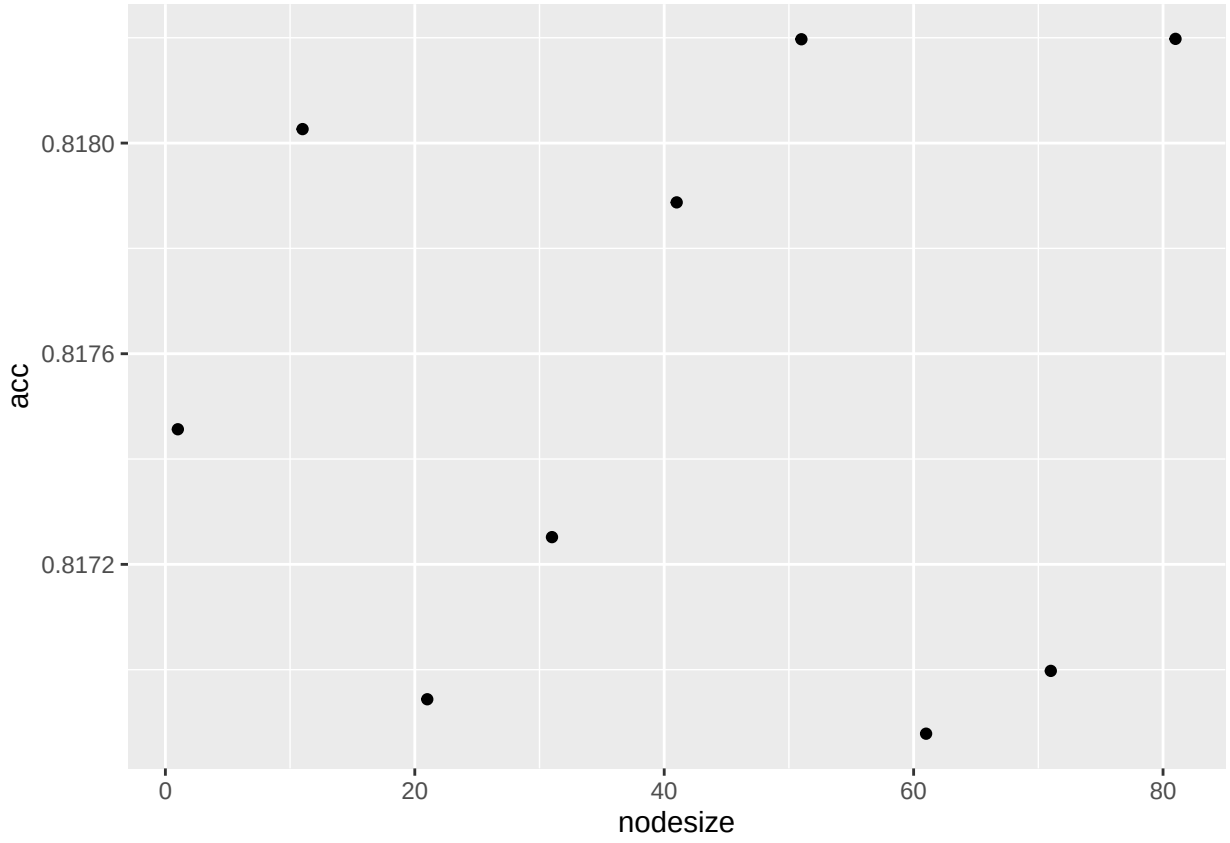
```
## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```

```
y_hat <- predict(train_rpart,adultpayclean_validation)
cm_rpart <- confusionMatrix(y_hat, as.factor(adultpayclean_validation$income  == "Above50K"))
accuracy_rpart <- confusionMatrix(y_hat, as.factor(adultpayclean_validation$income  == "Above50K"))$ove

sensitivity_rpart <- cm_rpart$byClass[["Sensitivity"]]
specificity_rpart <- cm_rpart$byClass[["Specificity"]]
prevalence_rpart <- cm_rpart$byClass[["Prevalence"]]

#random forest
set.seed(2008)
train_rf <- randomForest(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship,

cm_rf <- confusionMatrix(predict(train_rf, adultpayclean_validation),
                         as.factor(adultpayclean_validation$income  == "Above50K"))
accuracy_rf <- confusionMatrix(predict(train_rf, adultpayclean_validation),
                as.factor(adultpayclean_validation$income  == "Above50K"))$overall["Accuracy"]

sensitivity_rf <- cm_rf$byClass[["Sensitivity"]]
specificity_rf <- cm_rf$byClass[["Specificity"]]
prevalence_rf <- cm_rf$byClass[["Prevalence"]]

#random forest with tuning
nodesize <- seq(1, 90, 10)
acc <- sapply(nodesize, function(ns){
  train(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationship, method = "rf", da
        tuneGrid = data.frame(mtry = 2),
        nodesize = ns)$results$Accuracy
})
```

```
## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```

```
## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```

```
qplot(nodesize, acc)
```




```
train_rf_2 <- randomForest(y ~ age + eduyears + sex + race + hoursperweek + maritalstatus + relationshi
                           nodesize = nodesize[which.max(acc)])

y_hat_rf2 <- predict(train_rf_2, adultpayclean_validation)

cm_rf2 <- confusionMatrix(predict(train_rf_2, adultpayclean_validation),
                          as.factor(adultpayclean_validation$income  == "Above50K"))

accuracy_rftune <- confusionMatrix(predict(train_rf_2, adultpayclean_validation),
                as.factor(adultpayclean_validation$income  == "Above50K"))$overall["Accuracy"]

sensitivity_rf2 <- cm_rf2$byClass[["Sensitivity"]]
specificity_rf2 <- cm_rf2$byClass[["Specificity"]]
prevalence_rf2 <- cm_rf2$byClass[["Prevalence"]]

# tabulate all the accuracy results with sensitivity and specificity
accuracy_results <- matrix( c("Plain old guess", round(accuracy_guess,5), round(sensitivity_guess,5), r
```

	Method	Accuracy	Sensitivity	Specificity	Prevalence
1.	Plain old guess	0.50034	0.48329	0.50591	0.24606
2.	linear model	0.81666	0.47911	0.92682	0.24606
3.	General linear model	0.8098	0.38858	0.94727	0.24606
4.	knn	0.80809	0.90364	0.51532	0.75394
5.	knn3	0.80843	0.90136	0.52368	0.75394
6.	knn tune	0.81151	0.90919	0.57043	0.75419
7.	rpart	0.82111	0.91	0.54875	0.75394
8.	rf	0.82351	0.91636	0.539	0.75394
9.	rf tune	0.82248	0.92318	0.51532	0.75394

```
                        "linear model", round(accuracy_lm,5),round(sensitivity_lm,5), round(specificit
                        "General linear model", round(accuracy_glm,5),round(sensitivity_glm,5), round
                        "knn", round(accuracy_knn,5),round(sensitivity_knn,5), round(specificity_knn,5
                        "knn3", round(accuracy_knn3,5),round(sensitivity_knn3,5), round(specificity_kn
                        "knn tune", round(accuracy_knntune,5),round(sensitivity_knntune,5), round(spec
                        "rpart", round(accuracy_rpart,5),round(sensitivity_rpart,5), round(specificity
                        "rf", round(accuracy_rf,5),round(sensitivity_rf,5), round(specificity_rf,5), r
                        "rf tune", round(accuracy_rftune,5),round(sensitivity_rf2,5), round(specificit
),
  nrow = 9, ncol=5, byrow=TRUE,
dimnames=list(c("1.","2.","3.","4.","5.","6.","7.","8.","9."),c("Method","Accuracy","Sensitivity","Speci

accuracy_results %>% knitr::kable() %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```