Developing models to predict cytotoxicity of metal nanoparticles

TXT

12/1/2021

Setup R packages for model development

After installing R and RStudio, open RStudio and run the following codes to install and load necessary packages for developing models and making web applications::

```
packages <- c("svDialogs", "data.table", "data.table", "openxlsx", "mlbench", "caret",
              "tools", "DT", "magrittr", "ggplot2", "nortest", "tseries", "stringr",
              "RcmdrMisc", "lmtest", "dplyr", "randomForest", "shiny", "shinydashboard")

install.packages(setdiff(packages, rownames(installed.packages())))

library(svDialogs)
library(data.table)
library(openxlsx)
library(mlbench)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(tools)
library(DT)
library(ggplot2)
library(car)
```

```
## Loading required package: carData
```

```
library(nortest)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(RcmdrMisc)
```

```
## Loading required package: sandwich
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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:dplyr':
##
##     combine
```

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

Developing models for predicting cytotoxicity of metal nanoparticles

Choose excel file containing dataset and read the dataset:

```
DataMetal <- read.xlsx("MetalESN.xlsx", sheet = 1, startRow = 1, colNames = TRUE,
                       rowNames = FALSE, detectDates = FALSE, skipEmptyRows = TRUE,
                       skipEmptyCols = TRUE, rows = NULL, cols = NULL,
                       check.names = FALSE, namedRegion = NULL,
                       na.strings = "NA", fillMergedCells = FALSE)

DataMetal <- select(DataMetal, c("Toxicity",
                                 "Dose",
                                 "Assay",
                                 "Time",
                                 "Species",
                                 "Cancer",
                                 "Cell_Tissue",
                                 "Cell_line",
                                 "SSA",
                                 "Zeta",
                                 "HSize",
                                 "CoreSize",
                                 "Coating",
                                 "Shape",
                                 "Metal"))
```

Split data into training and test set (70/30)

```
set.seed(1991)
split_size <- floor(0.70 * nrow(DataMetal))
in_rows <- sample(c(1:nrow(DataMetal)), size = split_size, replace = FALSE)
train <- DataMetal[in_rows, ]
test <- DataMetal[-in_rows, ]
```

Train Random Forest model:

```
train.control <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
RFmodel <- train(Toxicity ~ ., data = train, method = "rf", ntree = 100, trControl = train.control)
print(RFmodel)
```

```
## Random Forest
##
## 1403 samples
##   14 predictor
##    2 classes: 'NON_TOXIC', 'TOXIC'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 1262, 1262, 1263, 1263, 1263, 1262, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##     2   0.8238213  0.1845892
##    68   0.9154577  0.7138967
##   134   0.9158772  0.7215603
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 134.
```

Use RFmodel to predict test set:

```
predictions <- RFmodel %>% predict(test); predictions_train <- RFmodel %>% predict(train)
```

Get confusion matrix and performance of model:

```
CMatrix <- confusionMatrix(predictions, as.factor(test$Toxicity))
Performance <- data.frame(Parameter = row.names(as.data.frame(CMatrix$byClass)),
                          Value = as.data.frame(CMatrix$byClass))
colnames(Performance) <- c("Parameters", "Values")
CMatrix
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  NON_TOXIC TOXIC
##   NON_TOXIC       480    25
##   TOXIC            14    83
##
##                Accuracy : 0.9352
##                  95% CI : (0.9125, 0.9535)
##     No Information Rate : 0.8206
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7709
##
##  Mcnemar's Test P-Value : 0.1093
##
##             Sensitivity : 0.9717
##             Specificity : 0.7685
##          Pos Pred Value : 0.9505
##          Neg Pred Value : 0.8557
##              Prevalence : 0.8206
##          Detection Rate : 0.7973
##    Detection Prevalence : 0.8389
##       Balanced Accuracy : 0.8701
##
##        'Positive' Class : NON_TOXIC
##
```

```
Performance
```

```
##                               Parameters    Values
## Sensitivity                  Sensitivity 0.9716599
## Specificity                  Specificity 0.7685185
## Pos Pred Value            Pos Pred Value 0.9504950
## Neg Pred Value            Neg Pred Value 0.8556701
## Precision                      Precision 0.9504950
## Recall                            Recall 0.9716599
```

```
## F1                                   F1 0.9609610
## Prevalence                     Prevalence 0.8205980
## Detection Rate             Detection Rate 0.7973422
## Detection Prevalence Detection Prevalence 0.8388704
## Balanced Accuracy       Balanced Accuracy 0.8700892
```

Save data and models for later use in web application:

```
save(RFmodel, file = "RFmodel.RData")
```