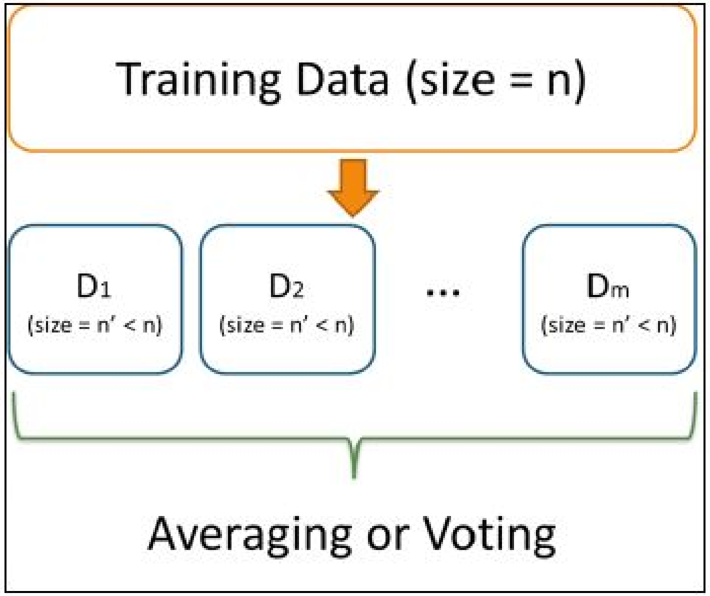
Machine Learning - 1100-ML0ENG (Ćwiczenia informatyczne Z-23/24)

Home > My courses > Machine Learning - 1100-ML0ENG (Ćwiczenia informatyczne Z-23/24) > Ensemble learning > Random forest

Random forest

Bagging (**B**ootstrap **agg**regat**ing**) is derived from the name Bootstrap aggregating.
The definition of bagging is as follows:

1. given **a training dataset of size n**,
2. bagging performs Bootstrap sampling (**random sampling with replacement**) and generates m new training sets, D_i each of size n.
3. finally, we can fit **m** Bootstrap samples to **m models** and combine the result by **averaging the output** (for regression) or **voting** (for classification)



Random forest

- Let us assume that we have a t**raining set containing N samples with M features (columns).**
- The process first performs **bootstrap sampling**, which samples N cases at random, with the replacement **as the training dataset of each single decision tree.**
- Next, in each node, **the process first randomly selects m variables** (where m << M), then finds the predictor variable that provides the best split among m variables.
- Next, the process grows the **full tree without pruning**.
- In the end, we can obtain the **predicted result of an example from each single tree**.
- As a result, we can get the prediction result by taking an average or weighted average (for regression) of an output or taking **a majority vote (for classification).**

Dataset biopsy

(Mastering Machine Learning) Dr. William H. Wolberg from the University of Wisconsin commissioned the Wisconsin Breast Cancer Data in 1990. His goal behind collecting the data was to identify whether **a tumor biopsy was malignant or benign**. His team collected the samples using Fine Needle Aspiration (FNA). If a physician identifies the tumor through examination or imaging an area of abnormal tissue, then the next step is to collect a biopsy. FNA is a relatively safe method of collecting the tissue, and complications are rare. Pathologists examine the biopsy and attempt to determine the diagnosis (malignant or benign).

The data frame is available in the R MASS package under the biopsy name.

This dataset consists of tissue samples from 699 patients. It is in a data frame with 11 variables, as follows:

- ID: Sample code number
- V1: Thickness
- V2: Uniformity of the cell size
- V3: Uniformity of the cell shape
- V4: Marginal adhesion
- V5: Single epithelial cell size
- V6: Bare nucleus (16 observations are missing)
- V7: Bland chromatin
- V8: Normal nucleolus
- V9: Mitosis
- class: Whether the tumor diagnosis is benign or malignant; this will be the outcome that we are trying to predict

```
#install.packages("MASS")
library(MASS)
data(biopsy)
str(biopsy)
biopsy <- biopsy[, -1] #delete ID
names(biopsy) <- c("thick", "u.size", "u.shape", "adhsn", "s.size", "nucl", "chrom",
"n.nuc", "mit", "diag") # change variable's name
summary(biopsy) #variable nucl has missing values, so we omit them.
biopsy.v2 <- na.omit(biopsy)
```

We create a **training set and test set.**

```
library(caTools)
set.seed(123)
split = sample.split(biopsy.v2$diag, SplitRatio = 0.7)
biop.train = subset(biopsy.v2, split == TRUE)
biop.test = subset(biopsy.v2, split == FALSE)
```

```
prop.table(table(biopsy.v2$diag))
prop.table(table(biop.train$diag))
```

The target variable is the diag column. We will build a classifier - a random forest, by this method we are going try to determine the value of the target variable based on the values recorded in the other columns.

randomForest package

We will use the **randomForest** package. The general syntax to create a random forest object is to use the **randomForest()** function and specify the formula and dataset as the two primary arguments.

- **mtry** - for regression, the default variable sample per tree iteration is $\frac{p}{3}$, and for classification, it is $\sqrt{p}$, where p is equal to the number of predictor variables in the data set. For larger datasets, in terms of p, you can tune the **mtry parameter**, which will determine the number of p sampled at each iteration.
- **ntree** - numbers of trees in the forest. Dafault value is 500

With this, let's build our forest and examine the results, as follows:

```
install.packages("randomForest")
library(randomForest)
set.seed(321)
rf.biop = randomForest(diag~., data=biop.train) # randomforest with default values
for ntree and mtry.
        # diag~. - formula for classification
rf.biop #the model of random forest
#values fo parameters in model

rf.biop$ntree
rf.biop$mtry
```

Now, we proceed to study the classifier behavior on the test set - **the predict function**. Then we build a confiusion matrix to compare the values we have in the dataset and the values we get using the classifier.

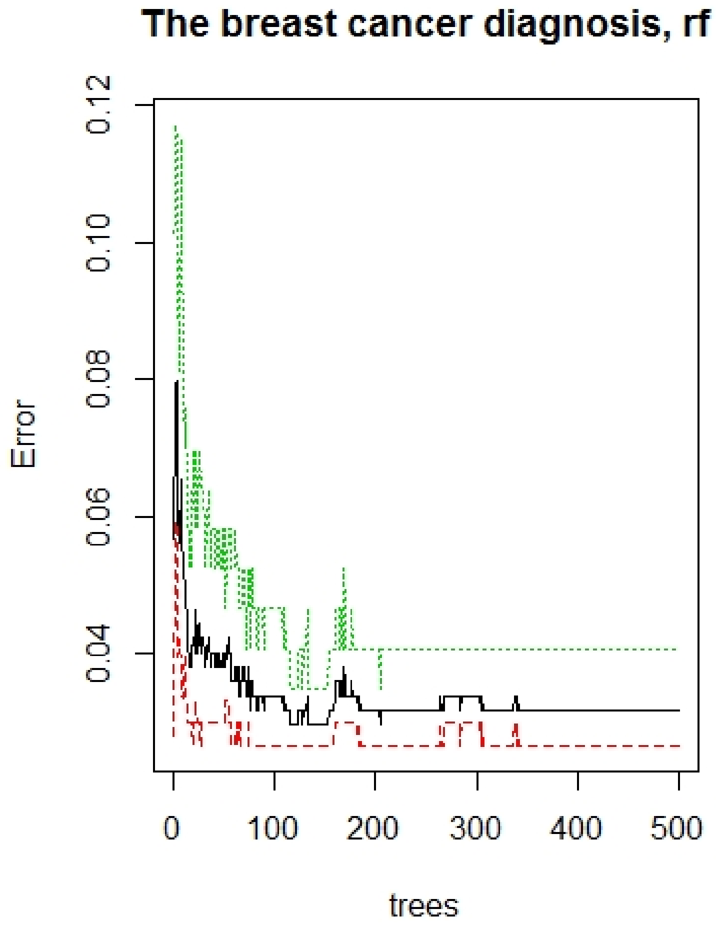
```
rf.biop.pred <- predict(rf.biop, newdata= biop.test)
table(rf.biop.pred, biop.test$diag)
```

```
> table(rf.biop.pred, biop.test$diag)

 rf.biop.pred   benign malignant
 benign            129         4
 malignant           4        68
```

We see four errors - our clasiffier says malignant in the dataset we have benign.

Let's plot the error (by trees)

```
plot(rf.biop, main="The breast cancer diagnosis, rf")
```



The chart shows that we don't need so many trees. We can build a smaller model for 200 trees. We can limit the number of trees

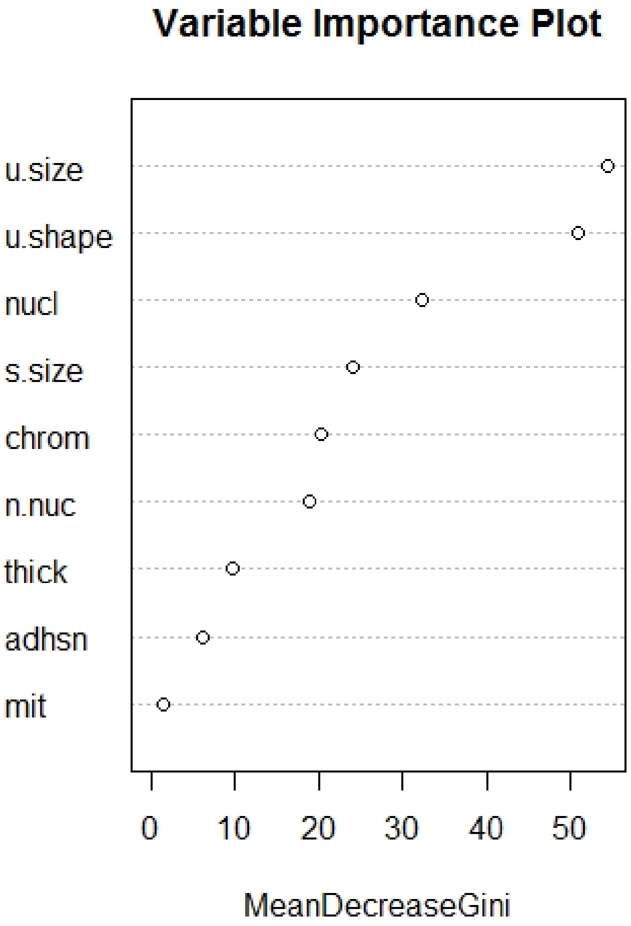
```
set.seed(321)
rf.biop.2 <-randomForest(diag~., data=biop.train, ntree=100)

rf.biop.pred.2 <- predict(rf.biop.2, newdata= biop.test)
table(rf.biop.pred.2, biop.test$diag)
```

```
-
```

Unfortunately, this did not improve the situation with classification errors.

Let's see another plot. You can produce a variable importance plot and corresponding list. The y-axis is a list of variables in descending order of importance and the x-axis is the percentage of improvement in MSE(minimum square error).

```
rf.biop$importance
varImpPlot(rf.biop,main="Variable Importance Plot")
```



You see the most important varialbles are **u.size, u.shape.**

Last modified: środa, 20 grudnia 2023, 9:25

Accessibility settings

Przetwarzanie danych osobowych

Platformą administruje Komisja ds. Doskonalenia Dydaktyki wraz z Centrum Informatyki Uniwersytetu Łódzkiego Więcej

Informacje na temat logowania

Na platformie jest wykorzystywana metoda logowania za pośrednictwem Centralnego Systemu Logowania.

Studentów i pracowników Uniwersytetu Łódzkiego obowiązuje nazwa użytkownika i hasło wykorzystywane podczas logowania się do systemu USOSweb.

Deklaracja dostępności