MS456/556 2021 Home assignment 1

Marks: out of 50 points.

Published: 26/02/2021

Submission deadline: Sunday 07/03/2021 by 8pm GMT

In this assignment, you are provided a dataset with 13 explanatory variables (attributes) in car insurance application. The task is to predict the *CLAIMFLAG* variable which has 2 levels describing whether the claim has been accepted with little/no change ('YES') or substantially revised by the insurer ('NO').

Variable	Description
KIDS	Number of kids of driver
AGE	Age of driver
INCOME	Income of driver (in EUR)
HOMEVAL	Value of home owned by driver (in EUR)
MARSTATUS	Marital status of driver
GENDER	Gender of driver
EDUCATION	Highest education degree of driver
TRAVTIME	Avg. travel time (hours/month)
CARUSE	Car use type
CARTYPE	Car type
CLMFREQ	Number of past claims
MVRPTS	Claim points
CARAGE	Age of the car

Rules:

- Feel free to talk to other members of the class in doing the homework. I am more concerned that you learn how to solve the problem than that you demonstrate that you solved it entirely on your own. You must, however, write down your solution yourself. If you fail to explain what any part of your code does if asked you will get 0 for this entire assignment.
- The assignment is to be submitted via Loop by the submission deadline (the submission is automatically time-stamped). Handwritten solutions are not allowed. Your submission should consist of a **single R script of up to 500 lines** (any excess will be disregarded) clearly highlighting the portions of code corresponding to individual task. The output **must also be copied inside the comments in the script** (immediately after the command) and should correspond to the R code.
- The R **script must compile without errors**! If some of the code instructions raise errors the corresponding task will be given 0 mark. Your script has to cover the whole of the task: from loading the data to processing it and reporting the output.

- You are strongly encouraged to use the libraries demonstrated at the tutorials. If your code uses substantially dissimilar functionalities you will automatically be invited to come and explain your script line by line.
- Carefully read the assignment and do only what you are asked to do. Any further code/results will not be considered and will not affect the final mark.
- Every time you use a function producing random output, like *randomForest, createFolds, train,* use ***set.seed(1234)*** to ensure that your results are reproducible.

Recommended libraries: shown in tutorials, i.e. *dplyr, caret, randomForest, e1071, rpart, rpart.plot, ada.*

Task 0. [data preparation]

Download from Loop the dataset *ha1_data.csv* and load the data into R from your local hard disk. You can use *read.csv* function to do this.

The dataset is randomised (*does not* require further reshuffling) and contains 7500 entries. Assign the first 80% to training data and the last 20% to test dataset.

NB: The dataset is real so don't expect to achieve overly high test accuracies if the following tasks.

Tasks 1. [8 points]

1.1 Train a default decision tree using *rpart* library. Report both train and test accuracies and kappas.

1.2 Plot the resulting tree with *rpart.plot*, and using it identify the smallest tree leaf. For this leaf explicitly formulate the associated rules.

1.3 Use *rpart.control* to find an alternative DT that improves the train accuracy by at least 3% (absolute value, e.g. from 76% to 79%). How does this modification affect the test performance? In terms of over/under-fitting comment if this modification appears to be better than the default DT.

Tasks 2. [18 points]

2.1 Construct a default random forest for the classification problem using *randomForest* function (from a library with the same name). Report train and test accuracy and kappa.

2.2 Establish the ranking (order) of importance of the explanatory variables used to estimate CLAIMFLAG. You can use *varImpPlot* function.

2.3 Establish which number of trees in the Random Forest delivers the optimal test performance, i.e. demonstrating the least of under/overfitting. Comment on your choice. From this perspective explain if the default *randomForest* setting performs well. [Use *seq(50,600,by=50)* for the number of trees in RF.]

2.4 Using the variable importance from 2.2 establish (and substantiate the claim) the highest number of input variables that one can omit (i.e., NOT use in the tree construction) in the classification task at hand if one is willing to have the test accuracy reduced by at most 5% (absolute value, e.g. from 76% to 71%) using random forests. Use as RF setting the optimal number of trees established in 2.3.

Tasks 3. [18 points]

3.1 Define folds for 10-fold cross-validation using *createFolds* function.

3.2 Find the best performing Boosted Trees classifier using '*ada'* method of *caret* library. To this end perform a grid search over *maxdepth* parameter (in range from 2 to 10) to find the best configuration. Fix *nu* to 0.1 and *iter* to 10. Use 10-fold cross-validation and the folds defined in 3.1. Report the test and train accuracies for the best configuration. Comment on which setting is best in terms of kappa and which in terms of accuracy.

3.3 Using the importances established in 2.2 identify (and substantiate the claim) the minimum number of input variables required to achieve the level of train accuracy within 3% of the best performance identified in 3.2 (absolute value of accuracy, e.g. from 76% to 73%) using Boosted Trees ('*ada'* method of *caret* library). To do so conduct a grid search fixing the *maxdepth* parameter identified in task 3.2, *nu* = 0.1, and try the values of parameter *iter* in the set *c(5,10,20)*.

Tasks 4. [6 points]

4.1 Comment on relative performance of methods in 1.3, 2.3 and 3.2.

4.2 Consider now that this is a real dataset that an insurance company works with. Would you recommend that the company collects more data or the available 7500 records are reasonable to address *CLAIMFLAG* prediction? Provide reasons and your conclusions about this dataset.