Stat 360 Project Rubric

Brad McNeney

Project overview

The project is to create an R package that implements the Multivariate Adaptive Regression Splines (MARS) algorithm described in: Jerome H. Friedman. "Multivariate Adaptive Regression Splines." Ann. Statist. 19 (1) 1 - 67, March, 1991. https://doi.org/10.1214/aos/1176347963.

The main function, `mars()`, takes a formula, data frame, and control object as input, fits the model by the forward and backward stepwise algorithms (Algorithms 2 and 3), and returns a `mars` object that contains the final least squares fit, along with a description of the basis functions from the final fit. You will also write `predict()`, `plot()`, `summary()`, and `print()` methods for `mars` objects. The `mars()` function and these four methods will be documented using `roxygen2` to produce R documentation files. Your package will include a test dataset that I will provide and unit tests of `mars()`, the `fwd_stepwise()`, `bwd_stepwise()` and `predict.mars()` functions that implement the forward and backward algorithms and the predict method. You will also include a package vignette that shows users how to use `mars()` and all four `mars` methods. In the vignette, include an analysis of one dataset that you find interesting (and that is not the test dataset). The dataset could be from another R package (easiest) or could be one that you have downloaded from another source and included as a package dataset.

The package will be made available in a `mars` directory within your group's project folder in the SFUStat360Projects GitHub repository. If your group's project folder is `MyGroup`, then your package should be in the `Projects/MyGroup/mars` folder of the `SFUStat360Projects` repository. You will also upload (i) a PDF of your vignette and (ii) a PDF of the package reference manual built with `devtools::build_manual()` to a Crowdmark assessment where they will be marked. The project is due at noon on Tuesday April 12.

Grading Scheme

Code on GitHub (25 marks)

At the project due date/time, all projects will be pulled from the SFUStat360Projects GitHub repository. The markers will copy the standard `testthat` suite from the `Exercises/ProjectTestfiles` folder of the SFUStat360 GitHub repository to your package's `tests` directory and will run both `devtools::test()` and `devtools::check()` on the package. The grading scheme for your code is as follows.

1. REAMDE (1 marks): A README.md file in your project's main folder (outside the `mars` pacakge folder) should include your group members' names and student numbers.
2. Working R package (11 marks):
   - (8 marks) The package should pass the unit tests of `mars()`, `fwd_stepwise()`, `bwd_stepwise()` and `predict.mars()`.
   - (1 mark) The package should include DESCRIPTION, LICENCE and NAMESPACE files, and data, data-raw, man, R, tests and vignettes directories
   - (2 marks) In addition to passing the unit tests and having the required structure, the package should pass `devtools::check()`.
3. `mars.R` (4 marks): The main `mars.R` file should include the `mars()` function and any others, such as `fwd_stepwise()`, that are called by `mars()`. Arrange your functions in a "top-down" manner, with

higher-level functions appearing first, followed by successive levels of lower-level functions. In addition to passing the unit tests (see part 1) you are graded on the following criteria:

- Data structures (2): The input data structures should be a formula, data and `mars.control` object. The output data structure is an S3 object of class `mars` that inherits from class `lm`.
- Readability (1): The steps and logic of your implementation should be clearly laid out. It should be easy for someone else in the class to read your code and understand what is going on.
- Efficiency (1): Take steps to avoid computational inefficiencies, such as copying large R objects.

4. Methods (9 marks): Include one file for each of the `plot()`, `predict()`, `print()` and `summary()` methods. The predict method will be tested with the unit tests. The criteria for the other three methods (plot, print and summary) are:

   - Informative (2 marks): The output should give useful information about the fitted model.
   - Familiar (1 marks): The method should look familiar to `lm()` users.

Documentation (25 marks)

By noon April 12 submit PDF files for the package vignette and the package reference manual to Crowdmark.

- To obtain the PDF of your vignette, go to the vignette folder and open the .Rmd file, knit it, open the resulting .html file in a browser, and use your browser's print feature to save the document as PDF.
- To obtain the PDF reference manual, use `devtools::build_manual()` to build a PDF that will be saved to the parent directory of your `mars` package directory.

Grades are assigned as follows.

1. Vignette (11 marks): The vignette should include an analysis of one dataset that you find interesting (and that is not the test dataset). There is some overlap between the vignette and the documentation of the `mars()` function. Think of the vignette as long-form documentation that **teaches** a user how to use **all** of the features of your package, rather than the terse documentation of the individual functions that is a **reference** for users who already know something about the package and just want a refresher or a quick-start on specific functions. The vignette should:

   - (2 marks) be logically organized for the goal of introducing your package and its features to a user.
   - (2 marks) give a description of the MARS algorithm
   - (1 mark) show the user how to prepare the inputs and call `mars()`
   - (4 marks) shown the user how to use all four methods on a `mars` object
   - (2 marks) use the most interesting data you can find. Trivial examples will get no marks.

2. Documentation for `mars()` (10 marks): The documentation for your `mars()` function must be generated from roxygen2 comments in your `mars.R` source file. Marks are allocated as follows:

   a. (1 marks) Description (brief) – a one- or two-line description of what the function does
   b. (1 mark) Usage – how to call the function
   c. (1 marks) Arguments – a list of arguments and their meaning
   d. (2 marks) Details – a precise and detailed description of what the function does
   e. (1 marks) Value – a description of the function's return value
   f. (1 mark) Author(s) – your name(s)
   g. (1 mark) References – a reference to the Friedman paper
   h. (1 marks) See Also – a brief description of the methods written for MARS objects
   i. (1 mark) Example – An example of how to use your function using the test dataset.

3. (4 marks) Documentation for the `predict()`, `plot()`, `summary()`, and `print()` methods will also be prepared as roxygen2 comments in each of the respective source files.