Homework 4 - Modern methods

Instructions

2022-09-02

0.0.1 Instructions

- Use the space inside of

```
::: {.solbox data-latex=""}
```

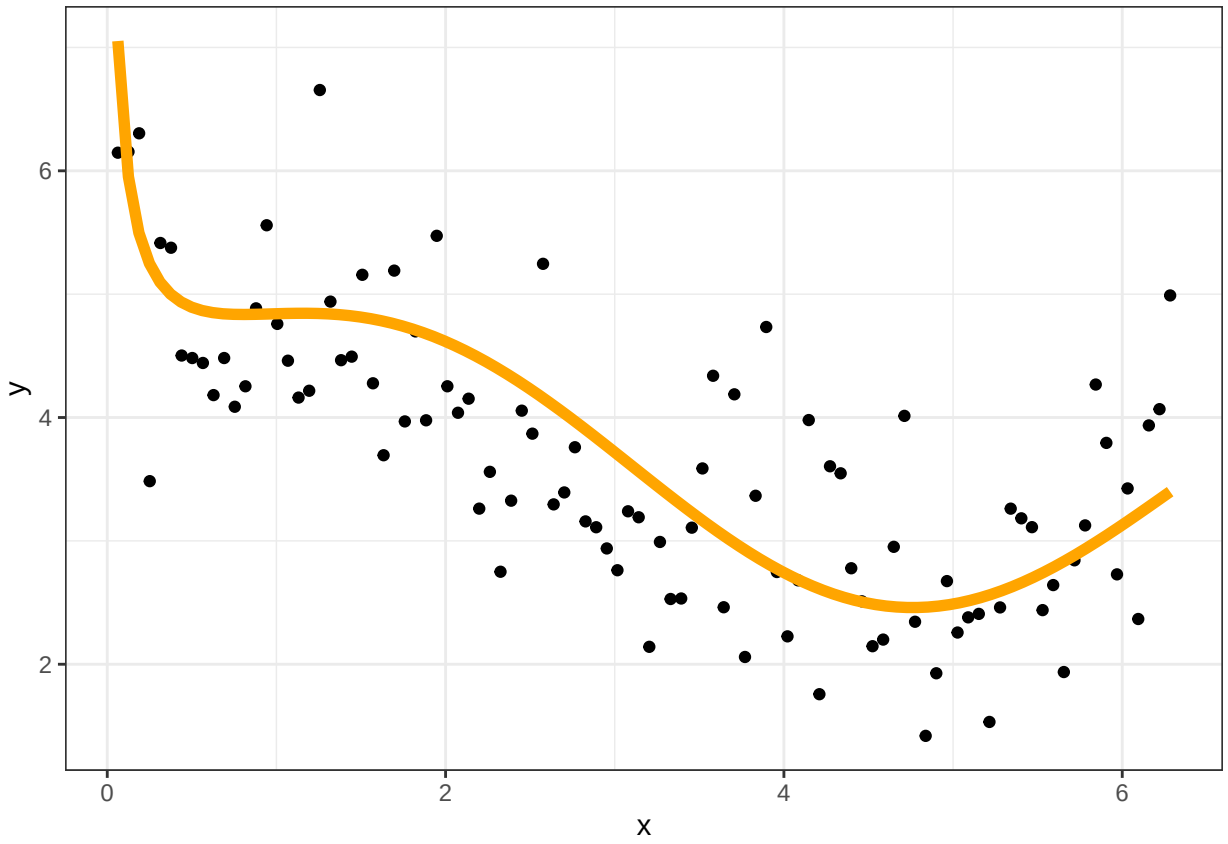
```
:::
```

to answer the following questions.

- Do not move this file or copy it and work elsewhere. Work in the same directory.
- Use a new branch named whatever you want. Create it now! Can't come up with something, try here. Make a small change, say by adding your name. Commit and push now!
- Try to Knit your file now. Are there errors? Fix them now, not at 11pm on the due date.
- There MUST be some text between `::: {.solbox}` and the next `:::` or this will fail to Knit.
- If your code or figures run off the .pdf, you'll lose 2 points automatically.

1 Bootstrapping

The code below loads some data and plots the data along with $E[y \mid X = x]$ (the true regression function). See `?wiggly_function` for the way it was created.


6
4
2
y
0
2
4
6
x

1. (2 points) Complete the function below. It has 3 arguments, (a) a data frame with `x` and `y` variables, (b) a `newdata` argument that expects a data frame containing at least `x`, (c) a switch that chooses between fitting a GAM or fitting a polynomial regression of degree 6. See the slides from Module 2 for help on GAMs, and polynomial regression. Both should include an intercept. Your function must return a vector with as many elements as there are rows in `newdata` with a prediction for each `x` value. Do not change the function signature. You may need to examine `?match.arg()`. You should make your function produce errors if invalid inputs are passed. This is easiest to do with `stop()` or `stopifnot()`. Try examining the documentation for those functions. Looking at the tests should help you determine what sorts of arguments to check against.

```
wiggly_estimator <- function(df, newdata = df, method = c("gam", "poly")) {



}
```

2. Use your wiggly estimator to produce predictions at the original `x` values with both the GAM and the degree 6 polynomial. Regenerate the above plot adding these curves colored `"purple"` and `"darkgreen"`.

```
#some code
```

3. Use the nonparametric bootstrap (resample the rows of the data) to produce 95% prediction intervals for `x0 = c(pi/2, pi, 3*pi/2)` using both the GAM and the polynomial model. Use `B=100`. Report your CIs in a table/matrix with the first column as `x0`, column 2 is the true `y` at `x0` columns 3 and 4 are the lower bound and upper bound for the GAM, columns 5 and 6 are the lower and upper bound for the polynomial. Round to 2 decimals.

```
x0 <- tibble(x = c(pi/2, pi, 3*pi/2))
B <- 100
```

4. Use the parametric bootstrap (resample the residuals from the fitted model) to produce 95% prediction intervals for `x0 = c(pi/2, pi, 3*pi/2)` using both the GAM and the polynomial model. Use `B=100`. Report your CIs in a table/matrix with the first column as `x0`, column 2 is the true `y` at `x0` columns 3 and 4 are the lower bound and upper bound for the GAM, columns 5 and 6 are the lower and upper bound for the polynomial. Round to 2 decimals.

```
#some code
```

5. Explain why both bootstrap methods are justifiable for this data.

Some text.

6. Which bootstrap intervals do you prefer and why?

Some text.

2 Bakeoff (4 points + potential bonus)

This exercise is based on the the TV show "The Great British Bakeoff". I am giving you information about bakers for the first 8 UK seasons. This data is derived from the package bakeoff by Alison Hill and Chester Ismay. You MAY NOT download this package (or use it). If you are unfamiliar with this show, I suggest you watch it! It's a great way to spend a few hours of your spare time.

A few other bits of important information. Again, easier if you just watch an episode or read Wikipedia, but nonetheless. Each episode consists of 3 "bakes": "signature challenge", "technical challenge", and "showstopper". In the technical, the bakers are ranked from best to worst. At the end of the episode, someone is awarded "star baker" (this week's winner), and someone goes home. Thus, the winners who make it to the last episode appear in more episodes. I have removed any information that depends on how long a baker lasts from the data. Also, in the first season, no one was awarded star baker. Because there is only one "overall winner" for the season, I have counted all the bakers that make the final episode as "winners". This means about a quarter of bakers are winners and the rest are losers.

> Your goal is to predict who will win and determine which features are useful for predicting the winners. Winners in this case are contestants involved in the final episode of the season (3 per season), rather than just the overall winner.

First, load the training data.

Notes:
There are 4 character variables that cannot be used for this analysis because of their type. Do not use them. (They could likely be used with careful processing, but I doubt very much that it is worth the effort. For this reason, I'm leaving them in the data in case you want them for the bonus. But also because real data often comes with useless cruft you should probably remove.) See also the `gbbakeoff.html` file for documentation.

```
data("bakeoff_train")
```

1. (1/2 points) In the data (after removing the 4 variables described in note above), there are 2 `numeric` variables which are entirely useless. They measure something which has nothing to do with winners and losers, and they are perfectly correlated with other variables. What are they?

Some text.

2. (1/2 points) There is one variable which is `numeric`, but should be treated as `categorical`. You must determine which, and treat it as 'categorical in all your models. Which is it?

Some text.

3. (1/2 points) Estimate a random forest to predict "winners". You must use 400 trees. Report the Out-of-Bag error rate of the full 400 tree ensemble. You likely need to set the `na.action` argument to `randomForest()` to `na.omit`.

```
ntrees <- 400
```

4. (1/2 points) Inspect the Random Forest object. It contains a number of different things. Show the confusion matrix (part of the object). Does the Forest do better for winners or losers for the training set?

```
# Some code
```

Some text.

5. (1 point) Describe the difference between Out-of-Bag error and In-Sample error.

Some text.

6. (1/2 points) Produce a Variable Importance plot. Which 2 predictors are most useful?

```
# Some code.
```

Some text.

7. (1/2 points) Using the same data, perform Bagging with 400 Trees. Report the confusion matrix. If your goal is "prediction accuracy" which method (RF or Bagging) should you use?

```
#some code.
```

Some text.

3 Bonus

You also have `bakeoff_test` in the package. This data is missing the response column. Your goal is to predict the response as accurately as possible.

The reward.

The student with the best score will receive 4 points on this homework submission (equivalent to 2% bonus on the final mark). The student with the second best score will receive 2 points. In the event of a first place tie (to 5 decimal places), all tied students will receive the first place bonus, but no second place will be awarded. In the event of a second place tie, all tied students will receive the second-place bonus.

Rules:

1. You must correctly upload one (and only 1) submission to the `bakeoff-bakeoff` repo. This must be done by a separate PR (from your usual HW submission). I would let you upload directly to `main`, but that would allow you to potentially delete submissions and sabotage your classmates.
2. Your submission must consist solely of a single `.rds` file. The name of the file must be `bake-ghusername.rds`. That file will contain 1 `R` object which is a vector of `TRUE` and `FALSE` with your predicted classes. It must be of type `logical`.
3. Suppose you have a vector `preds = c(TRUE, FALSE)`. You can achieve the above with the code `saveRDS(preds, "bake-ghusername.rds")`.
4. Neither I nor the TAs will provide **any** assistance. This means no questions on Slack or in office hours about approaches or methods for the bonus will be answered. You may use any methods you like. We will not resolve your Github issues on this repo nor will any late submissions be allowed. If your data is in the wrong format, you are ineligible.
5. I have taken precautions to avoid cheating. If cheating is suspected, you will be disqualified with no recourse (i.e., don't come and argue with me).
6. Have fun.