Homework 6

Due: 12:00 pm (noon), November 24, 2020

Here are some general guidelines.

Do not include your name on your write-up, since these will be peer-graded anonymously.

Do not include your raw R code in your write-up unless we explicitly ask for it. You will submit your R script as a separate document to the write-up itself. In Canvas, you will see actually *two* assignments corresponding to homework 5: one for the write-up, and one for the R script. Your write-up is what get's graded, but your R scripts must also be submitted along with the homework, by the same deadline, for the purpose of audits and ensuring compliance with course policy regarding academic integrity. If you do not submit your R script, you will not receive credit for the homework.

If you use tables or figures, make sure they are formatted professionally. Figures and tables should have informative captions. Numbers should be rounded to a sensible number of digits (you're at UT and therefore a smart cookie; use your judgment for what's sensible). Rows and columns in tables should line up correctly, and tables shouldn't merely be copied and pasted in Courier (or similar) directly from the R output.

Format your answers in the same way we've learned to do on previous homeworks, with four sections: 1) Questions; 2) Approach; 3) Results; 4) Conclusions.

Problem 1

This question considers data on sales volume, price, and advertisting display activity for packages of Borden sliced cheese, available in "cheese.csv". You've looked a slice of this data before (for Kroger's in the DFW area). You'll now examine the full data set. For each of 88 stores (store) in different US cities, we have repeated observations of the weekly sales volume (vol, in terms of packages sold), unit price (price), and whether the product was advertised with an in-store display during that week (disp = 1 for display). Altogether there are 5,555 observations in the data set.

The goal of this analysis is to understand consumer behavior for Borden slice cheese, by characterizing the price elasticity of demand for this market. Remember back to our milk sales-versus-price data: a typical model for price elasticity of demand is of the form $Q = KP^{\beta}$, where Q is quantity sold, P is price, K is a constant, and β is the elasticity—that is, the relative percentage change in sales as price changes by 1%. You should recall how to use linear least squares to fit such a model.

Build a model for Q (sales volume) in terms of price (P), store-level dummy variables, and a dummy variable for whether or not there was a display for cheese.

Use this model to answer the following questions, quoting appropriate confidence intervals:

- What is the price elasticity of demand for Borden sliced cheese in no-display weeks? Interpret this number in a single sentence (i.e. "When price of cheese goes up by 1%. . . ").
- Does price elasticity for Borden cheese appear to be changed by the presence of in-store display? (Hint: remember about interaction terms in models with numerical and categorical predictors.) Can you think of a possible economic explanation for your result here?
- What price should Kroger's in Dallas/Ft.~Worth charge for cheese in no-display weeks if their goal is to optimize gross profit?
- Adjusting for store-level differences and differences in price from week to week, how much higher or lower in percentage terms do sales seem to be in display weeks versus non-display weeks, on average across all stores?

Problem 2

The files `hotels_train.csv` and `hotels_test.csv` contain data on tens of thousands of hotel stays from a major U.S.-based hotel chain. The goal of this problem is simple: to use linear regression to build a machine-learning model for predicting whether a hotel booking will have children on it.

Why would that be important? For an equally simple reason: when booking a hotel stay on a website, parents often enter the reservation exclusively for themselves and forget to include their children on the form. Obviously, the hotel isn't going to turn parents away from their room if they neglected to mention that their children would be staying with them. But **not** knowing about those children does, at least in the aggregate, prevent the hotel from making accurate forecasts of resource utilization. So if, for example, you could use the *other* features associated with a booking to forecast that a bunch of kids were going to show up unannounced, you might know to order more chicken nuggets for the restaurant and less tequila for the bar. (Or maybe more tequila, depending on how frazzled the parents who stay at your hotel tend to be.) In any event, as a hotel operator, if you can forecast the arrival of those kids a bit better, you can be just a bit more efficient, operationally speaking. This is an excellent use case for an ML model: a piece of software that can scan the bookings for the week ahead and produce an estimate for how likely each one is to have a "hidden" child on it.

The target variable of interest is `children`: a dummy variable for whether the booking has children on it. All other variables in the data set can be used to predict the `children` variable.

Please compare the out-of-sample performance (measuring using RMSE) of the following four models:

1. a small model that uses only the `market_segment`, `adults`, `customer_type`, and `is_repeated_guest` variables as features.

2. a big model that uses all the possible predictors *except* the `arrival_date` variable (main effects only).

3. a huge model that uses all the possible predictors *except* the `arrival_date` variable, along with all their possible pairwise interactions.

4. the big model (model 2 on this list), with one additional "engineered" feature: the month of the year, based on the `arrival_date` variable. (Remember our use of the `lubridate` package in R to do this kind of feature engineering with dates.)

Use the data in `hotels_train.csv` to fit the models. Use the data in `hotels_test.csv` to calculate out-of-sample RMSE.

Notes and requirements:

- You don't need to report fitted model coefficients in your Results section. Really all your Results section needs to contain is a table with four rows (one for each model) and two columns (one for training-set RMSE, the other for test-set RMSE). Please report the RMSE numbers to four decimal places. Give the table an informative caption that describes what the table shows. Your Conclusions section should also be quite short—essentially just a recommendation about which model to use for predicting "hidden" children on hotel bookings.

- It may take awhile for the huge model to fit on your machine. Be patient.

Problem 3

Go read the article "One match to go!", by Spiegelhalter and Ng, linked from our Canvas site. In this article, the authors describe how they formulated an approach for predicting the probability of different outcomes for soccer matches based on "attack strength" and "defense weakness." It is better than the simple approach we took in class, though probably not as good as what actual bookmakers (i.e. in Las Vegas) use.

Now go download the data from the 2018-19 English Premiere League soccer season. (We'll use the data from two seasons ago because the 2019-20 season was disturbed by the pandemic, while the 20-21 season is only just getting started.) These are in the files "epl_2018-19_away.csv" and "epl_2018-19_home.csv", which give the home and away performance for all 20 teams. These two files allow you to replicate the analysis described in the "One Match to Go!" article. Specifically, the columns of interest in each file are "GF" and "GA", which are "Goals For" and "Goals Against," respectively. So, for example, in the "epl_2018-19_away.csv" you'll notice that Manchester City has 38 for GF and 11 for GA. That means in their away games that season, Manchester City scored 38 goals and allowed 11 goals by their opponents. (Note: this data were [downloaded from here].(http://www.soccerstats.com/latest.asp?league=england)

Replicate Spiegelhalter and Ng's approach using the 2018-19 data to answer the following two questions:

1. What is your estimated probability distribution of win/lose/draw results for a match between Liverpool (home) and Tottenham (away)?
2. What about Manchester City (home) versus Arsenal (away)?

Notes and requirements:

- You might find this easiest to do in Excel or a similar spreadsheet program (Google Sheets, Numbers, etc), although you can certainly use R if you want. If you use Excel or similar, you'll submit your spreadsheet and/or Google Sheets link as your upload for Problem 3, instead of an R script. You'll see a separate "Problem 3 work files" submission link for this homework.
- Spiegelhalter and Ng did the calculations for the whole League, but you certainly don't need to; these two games will suffice.
- Your write-up doesn't need the break down the results by all possible game scores (1-0, 1-1, etc); just summarize the probability of a draw or win for each team. The one exception is that, in your "Approach" section, you should include an explicit math formula that shows how you calculated the probability of a 2-1 victory for the home team in each of the two games. This formula will serve as an example of your approach for calculating all the necessary probabilities. You can insert an equation directly into the document via Word's Equation Editor or similar, or you can simply include a handwritten equation that you took a photo of.
- Summarize the Spiegelhalter and Ng approach in your own words in your Approach section of your write-up. It's fine to assume independence between the teams' scores but state this assumption in the Approach section.
- Don't get confused by the "Pts" (points) column. This isn't goals. "Points" are how the league crowns a winner, with a team getting 3 points for a win and 1 point for a draw.
- GP means "games played."

Problem 4

(Note: this problem won't follow the "Questions/Approach/Results/Conclusions" outline. See below for formatting requirements.)

Suppose you have $10,000 dollars to invest, and that you are offered the following wager, based on the outcome of a "biased" coin flip that comes up heads with probability $p = 0.52$. You get to choose what fraction c of your total wealth of $10,000 to wager on the outcome of the bet. You win the bet if the coin comes up heads. So:

- With probability $p = 0.52$, you will win the bet and therefore gain $c \cdot \$10,000$.
- With probability $1 - p = 0.48$, you will lose the bet, and therefore lose $c \cdot \$10,000$.

Since the coin is in your favor, it sounds like a good bet, right?

Part A

Suppose that you decide to risk $c = 0.10$ (i.e.~10%) of your wealth on this bet, and that you repeat the bet over and over again. After every single round of betting, you decide to risk the same fraction $c = 0.1$ of your current wealth on the next bet. In other words, if you have w_t dollars after round t of betting, you place $0.1 \cdot w_t$ dollars on the next round's wager. If you win, your new wealth will be $w_{t+1} = (1 + c) \cdot w_t$. If you lose, your new wealth will be $w_{t+1} = (1 - c) \cdot w_t$. Thus the ending point round $t = 1$ becomes the starting point for round $t = 2$, and so on.

Simulate 10,000 rounds of this bet.[1] What happens after 10,000 rounds of betting? Are you rich or broke? Run the simulation four times, in order to convince yourself of what happens here. Do you find this surprising?

To code up this simulation, we recommend that you build on our example R script from the walkthrough on Monte Carlo for sequential events. There are many ways to simulate the outcome of each bet in R code; for example, you might consider a combination of `ifelse` and `rbinom` (to simulate a binomial outcome for each round of the bet).

For this part, all you need to turn in is a single page with four plots, one each for your four simulated trajectories of wealth w_t over every round from $t = 1$ to $t = 10000$ (the betting round t should be on the x-axis). Make these four plots fits on a single page, and give the panel of four plots a caption explaining what seems to happen (i.e. whether you tend to get rich or go broke). Note: your panel of four figures will look like the output of `facet_wrap`, but you'll probably find it easier to just copy the four plots individually and paste them on a single page in your write-up.

Part B

Now repeat the simulation—but this time, you should risk only 0.5% of your current wealth (that is, $c = 0.005$, or 1 part in 200) at every round of betting. As before, plot four trajectories of simulated wealth w_t at every step from 1 to 10,000. Now what happens after 10,000 rounds? Are you rich or broke? Again, turn in a page with four plots for your four simulated trajectories, with a single caption for the whole panel.

Part C

Experiment with your Monte Carlo simulation to find a value of c that you like best in order to maximize the long-term growth of your portfolio. What value of c seems to be the best? As above, plot four trajectories of simulated wealth under your chosen value of c (all four on a single page). In your caption for this panel of figures, explain how you judged what value of c looked best to you.

[1] Why 10,000? Because that's roughly the number of trading days over a 40-year period of investment. The idea of this problem is to simulate the career of a trader who tries to make favorable trades every day.