Predictive Modeling with Sports Data

Homework 8 (Mini-Project)

Unless stated otherwise, justify any answers you give. You can work in groups, but each student must write their own solution based on their own understanding of the problem. Do not share any code.

When uploading your homework to Gradescope you will have to select the relevant pages for each question. Please submit each problem on a separate page (i.e., 1a and 1b can be on the same page but 1 and 2 must be on different pages). We understand that this may be cumbersome but this is the best way for the grading team to grade your homework assignments and provide feedback in a timely manner. Failure to adhere to these guidelines may result in a loss of points. Note that it may take some time to select the pages for your submission. Please plan accordingly. We suggest uploading your assignment at least 30 minutes before the deadline so you will have ample time to select the correct pages for your submission.

Make sure your answers to each problem are clearly stated in the submitted PDF. Do not include code in your submitted PDF. Any important figures, results, plots, and tables generated by your code should be extracted and inserted into the PDF in a visually appealing way. Think of the PDF as a presentation you are making based on the results you uncovered in your analysis. Your submitted PDF should be self-contained: the graders should not have to look through your code to find the solution.

Any code (Jupyter Notebooks and other source files) used to compute your answers should also be uploaded to Gradescope. If your code needs any special configuration in order to run, please include a `readme`. If the problem requires you to train and test a model, the training, validation, and testing code should all be submitted. If necessary, please indicate in your `readme` file if running your code requires special hardware (like a GPU, or 64GB+ of ram), a compilation step (if you decide to use Cython), or a signficant amount of time (longer than 5 minutes).

All of the questions in this homework will use the data in `xc.csv`. Throughout, restrict to data from flat races (where `race_type` is '`flat_race`'). In question 4, you can optionally try to extract information from the `handicap` and `hurdle` races, but it is not required (and may be difficult). For every fit (including both the multinomial logit and linear models), and in the test sets, restrict to races with at least 4 dogs, where every dog has ran in at least 3 strictly prior flat races, and every dog has run at least one flat race in the past 90 days. The column meanings are defined in the Live Lecture 11 slides.

1. (Baseline Model) In this question we build a basic model for predicting race winners. Construct a feature `avg_mmps` containing the average value of `mmps` from all strictly prior races for that dog. The definition of `mmps` is described in Live Lecture 11. You

can directly use the parameters defining mmps from the lecture. Fit a conditional multinomial logit model of the form

$$\texttt{twinner} \sim \texttt{avg\_mmps}.$$

Here twinner is equal to winner for races with a unique winner, and is a randomly chosen winner in the other cases (see the conditional logit model notebook for code that creates the twinner column, and for the mlogit function that fits a conditional multinomial logit model).

(a) Fit the above model on races between July 1st, 2019 and January 31st, 2020, and report your coefficients.

(b) Report your out-of-sample Brier score using the races on and after February 1st, 2020. This is computed using all of your forecasted probabilities, the twinner column, and the Brier score loss function from sklearn.

(c) Submit your results to this problem (i.e., 1 only) in a single PDF on gradescope (listed under HW 8 Check-in - 1). You will also resubmit your solutions to this problem when you submit the full miniproject (listed under HW 8).

2. (Building a Speed Model) In this question we will build a linear model to better predict dog speeds in upcoming races. The outputs of this model can then be used as a feature in our multinomial logit models.

   (a) Fit a linear model of the form

   $$\texttt{mmps} \sim \texttt{mmps\_ema}$$

   where mmps is the modified speed computed for a given dog in the current race, and mmps_ema is an exponentially weighted moving average of mmps using data for that dog from strictly prior races. Your decay length can either be specified in races or days, and you must state whichever one you choose. [Hint: If your EMA is specified in days, you can use the times parameter to ewm in pandas version 1.1 and later.]

      i. Fit the above model on races between July 1st, 2019 and November 30th, 2019, and report your coefficients.

      ii. Report your out-of-sample average square loss (for predicting mmps) using the races between December 1st, 2019, and January 31st, 2020, inclusive.

   (b) Improve your mmps prediction model in the preceding part by also incorporating the stadium_id. [Hints: Treat stadium_id like a categorical variable, but also remember to decay each indicator variable using the same EMA length as used in the preceding part, and incorporate the stadium_id of the current race.]

      i. Fit the above model on races between July 1st, 2019 and November 30th, 2019, and report your coefficients.

ii. Report your out-of-sample average square loss (for predicting `mmps`) using the races between December 1st, 2019, and January 31st, 2020, inclusive.

3. (Incorporating Comments) The `comment` column of our data includes useful information about what events happened to each dog during the course of the race. In this question we will incorporate the comment information into the `mmps` prediction model we built in the previous part.

   Create indicator variables for some of the potential events that can happen to each dog during the race (these indicator variables are defined by finding substrings present in each comment; see `race_comments.pdf` for more info). Add these indicators to the `mmps` prediction model from the previous question that included stadium effects. Remember to decay your indicator variables using the same EMA length as the other features.

   (a) Fit the above model on races between July 1st, 2019 and November 30th, 2019, and report your coefficients.

   (b) Report your out-of-sample average square loss (for predicting `mmps`) using the races between December 1st, 2019, and January 31st, 2020, inclusive.

4. (Improving the Baseline) In this final problem, we improve on our baseline model from the first question.

   (a) Build an improved multinomial logit model for `twinner` by adding the forecasts of our `mmps` prediction model as a feature.

      i. Fit the above model on races strictly before February 1st, 2020, and report your coefficients.

      ii. Report your out-of-sample Brier score using the races on and after February 1st, 2020. This is computed using all of your forecasted probabilities, the `twinner` column, and the Brier score loss function from `sklearn`.

   (b) Improve your model from the previous part in some way. You can do this by improving your `mmps` prediction model, or by adding features to the multinomial logit model. Note that you can use `stadium_id`, `kg`, `distance_m`, `race_grade`, and `box` from the **current race**, and `going`, `decimal_price` from **strictly prior races** in your fits.

      i. Fit the above model on races strictly before February 1st, 2020, and report your coefficients.

      ii. Report your out-of-sample Brier score using the races on and after February 1st, 2020. This is computed using all of your forecasted probabilities, the `twinner` column, and the Brier score loss function from `sklearn`.

   (c) Take your final model from the previous part, and fit a combined Benter-style model (i.e., use the logits of your forecast, and the logit of the market implied probabilities as the two features in a conditional multinomial logit model).

i. Fit the above model on races between July 1st, 2019 and January 31st, 2020, and report your coefficients.

ii. Report your out-of-sample Brier score using the races on and after February 1st, 2020. This is computed using all of your forecasted probabilities, the `twinner` column, and the Brier score loss function from `sklearn`.