Mini Project
MSiA401
Due Wed, Dec 7 at noon

Business situation: You have data from a German book company. The company would like to have a machine learning model to predict how much a customer will spend it sent an offer based on previous purchase history.

Data: On a certain date the book company sent an offer to some of its customers and then observed the responses to this offer. I have already done some feature engineering for you and the file `bookall.csv` has one row for each customer and different possible predictor variables. You should find 16,781 customers with a header line and 68 variables. Here is a description of the variables in order of the columns:

- `id`: uniquely identifies a customer. **Do not use this in your models.**
- `train`: takes the value 1 for the **training** set a 0 for the **test** set. Only use the training data to estimate your models! Of these, 5551 are training cases and 11,230 are test cases.
- `target`: the amount spent by the customer in response to the offer. This is the dependent variable of your analysis.
- `tof`: **time on file**, i.e., time since the first order
- `r`: **recency**, i.e., time since most recent order
- `fitem`: **item frequency** (number) of previous items
- `ford`: **order frequency** (number) of previous orders
- `m`: **monetary**, the total amount spent on previous orders
- `f1`, . . . , `f99`: frequency of **items** from some **category number** in the past. You will probably not need to know this, but if you are interested, the the category numbers are as follows: 1=fiction; 3=classics; 5=cartoons; 6=legends; 7=philosophy; 8=religion; 9=psychology; 10=linguistics; 12=art; 14=music; 17=art reprints; 19=history; 20=contemporary history; 21=economy; 22=politics; 23=science; 26=computer science; 27=traffic, railroads; 30=maps; 31=travel guides; 35=health; 36=cooking; 37=learning; 38=games and riddles; 39=sports; 40=hobby; 41=nature/animals/plants; 44=encyclopedia; 50=videos/DVDs; 99=non-books

- `m1`, ..., `m99`: monetary value spend on items from each category in the past.

Goal: suppose the company will use this model to select between 20% and 40% of the customers to receive an offer. They care about **test set** (1) percentage of responders (i.e., precision), (2) total amount of revenue generated from those contacted, (3) ROC/AUC, and (4) F_1.

Modeling task: Build a machine learning model to accomplish the goal. Here are some hints:

- I have intentionally given you a fairly small training set (massive data sets are easier!). You will probably need to do some sort of selection and/or shrinkage to avoid overfitting.

- You may want to add appropriate transformations to the model. Transformations could be simple such as taking a log of a variable, or could be more complicated combinations of two or more predictor variables.

- I expect to see several models with comparisons on the test-set metrics. For example, you could compare lasso, ridge, stepwise and the two-step model.

- Tell me what model(s) work(s) best along with what (transformed) predictors are in the model.

- You have a class imbalance problem. See here for some approaches to dealing with the problem. I don't expect you to try everything. Perhaps just try random oversampling. Answer this question: does oversampling buyers improve the model performance on the test set?

- Reflect briefly on which "modeling techniques" make a substantial difference, e.g., do transformations of y and/or x's make a big difference? Does the choice of the dependent variable matter (targdol, binary buy or not, two-step)? Does the method of selection and/or shrinkage matter much?