IEOR 242: Applications in Data Analysis, Fall 2019

Homework Assignment #4

October 28, 2019

Problem 1: Predicting Useful Questions on Stack Overflow (100 points)

Stack Overflow is a very popular question-and-answer website, featuring questions on many different computer programming topics. (The Stack Exchange Network also features many different spin-off sites on a wide variety of topics.) Users of Stack Overflow can post questions and also provide answers to questions. Users who have earned certain privileges can also vote on the quality/usefulness/helpfulness of both questions and answers. For each question (or answer), privileged users who have read the question can elect to provide either an "upvote" or a "downvote" for the question. An upvote constitutes an endorsement of the question's usefulness for other users, whereas a downvote signals that the question is likely to be especially not useful (or has been sloppily written, or does not show enough research effort, etc.). The "score" of a question is equal to the total number of upvotes for that question minus the total number of downvotes.

In this problem, you will work with question data from Stack Overflow, made available as part of the Stack Exchange Network "data dump".[1] The particular dataset that you will work with contains questions about the ggplot2 package in the R programming language, which were posted between January 2016 and September 2017.[2] The dataset is in the file `ggplot2questions2016_17.csv`. There are 7,468 observations, with each observation recording text data associated with both the title and the body of the associated question and also the score of the question. Table 1 describes the dataset in further detail.

After a question is posted, it would be beneficial for Stack Overflow to get an immediate sense of whether the question is useful or not. With such knowledge, the website could automatically promote questions that are believed to be useful to the top of the page. With this goal in mind, in this problem you will build models to predict whether or not a ggplot2 question is useful, based on the title and body text data associated with each question. To be concrete, let us say that **a question is useful if its score is greater than or equal to one.**

As compared to previous homework exercises in this course, this exercise is purposefully open ended. Therefore, your submission should be in the form of a brief report, with one section dedicated to each of the parts (a), (b), and (c) outlined below.

[1] https://archive.org/details/stackexchange

[2] These questions have all been tagged with both the "R" tag and the "ggplot2" tag – see http://stackoverflow.com/questions/tagged/r+ggplot2

Table 1: Description of the dataset `ggplot2questions2016_17.csv`.

Variable	Description
`Title`	Title text of the question
`Body`	Main body text of the question (in html format)
`Score`	Score of the question, equal to total number of upvotes minus total number of downvotes

a) (30 points) Start by cleaning up the dataset to get it into a form where we can apply statistical learning methods to predict our dependent variable of interest. Please briefly describe each step that you took to clean and process the data. Here are some suggestions:

i) Remember to set **`stringsAsFactors = FALSE`** when loading text data from a .csv file.

ii) The R package **`tm.plugin.webmining`** is useful for dealing with html text. To use this package, you will need to have the Java Development Kit (JDK) installed, which can be downloaded here. If you are using macOS (or possibly Linux/Unix), then after installing the JDK, it may also help to run the following command in the terminal:

```
sudo R CMD javareconf
```

After running the above command, you should then restart RStudio and try to load the **`tm.plugin.webmining`** library.

iii) Note that the function **`extractHTMLStrip`** in the previously installed package may be used to remove html tags. (Note also that **`extractHTMLStrip`** may or may not be currently working well with the **`tm_map`** function; therefore you may need to write a manual for loop to loop over your corpus in this case.)

iv) You should inspect one or more example questions during your cleaning process to make sure that all of your cleaning transformations are working smoothly. You may also need to write a custom transformation function or two; the **`gsub`** function in R is useful for this.

v) You should process the title text and the body text separately, and then join the resulting independent variables together. It's good practice in R to make sure that each independent variable in your final data frame has a unique name which is syntactically valid.

vi) Think carefully about the tradeoffs involved in selecting the number of independent variables in your final dataset and use your best judgment.

b) (40 points) Now split your processed data into a training set and a test set. Use 70% of the data as training data and be sure to split the data in such a way that keeps the relative amount of useful questions roughly the same in each set (we have used two functions in R for splitting data, only one of them is appropriate here). Use your analytics skills to build the best model that you can for predicting useful questions. You are free to try whichever approaches

you like, but you should try at least **four** distinct methods that have been discussed in class (e.g., Logistic Regression, LDA, CART, Random Forests, Boosting) and you should evaluate no more than a handful (i.e., 5-10) of candidate models on the test set. For Random Forests and Boosting, it is OK to not cross validate all of the parameters if it takes too much time on your computer – instead you may use some reasonable values of these parameters and/or only cross validate some of them.

Report on the details of your training procedures and final comparisons on the test set. Use your best judgment to choose a final model and explain your choice. Use the **bootstrap** to asses the performance of your final model in a way that properly reports on the variability of the relevant performance metrics (accuracy, TPR, and FPR).

c) (30 points) Now, let us consider how Stack Overflow might use your model. In particular, consider the following scenario. When a user navigates to the page showing ggplot2 questions, they are automatically presented with the 15 most recently submitted questions, in order of most recent first (this is not necessarily true in reality, but let's pretend that it is). Suppose that Stack Overflow would like to keep these same 15 most recently submitted questions on this page, but they would like to rearrange the order of the questions so as to show the most useful questions at the top of the page. Suppose further that Stack Overflow believes that most users are extremely impatient and will only pay attention to the single question at the very top of the page, and therefore they would like to **maximize the probability that the top question is useful**.

i) Think about how to select a model, among many different models, to best accomplish the goal of maximizing the probability that the top question is useful. Comment on the precise criteria that you would use (e.g., "I would select a model with the highest accuracy" or "I would select a model with the highest TPR", etc.) and note that your answer may involve multiple performance metrics if you wish. Explain your response.

ii) Revisiting the models that you have built in part (*b*), can you identify a specific model that best accomplishes the goal? (Note that this model may be different from the final model you selected in part (*b*), and you are allowed to do some re-training in addition to re-evaluating your models.) How much does the model you selected improve upon Stack Overflow's current approach of showing the most recent posts first (described above)? In particular, use the results of your model on the test set to give a precise numerical estimate of the increase in the probability that the top question is useful. If you like, you may use a back-of-the-envelope "on average" style of analysis as part of your reasoning to answer this question.