Jiangxue Han (jh6rg), Nick Bruno (nhb3zf), Beni Shpringer (bs2ux)

SYS 6018: Kaggle Blog Competition

The main objective of this competition was to predict blogger ages based on the contents of their blog posts. The reason for looking at this was not only to develop our text mining skills, but also because there are some implications that could be useful overall. For instance, a marketer might want to be able to know how old a blogger or tweeter is based on what they are writing in order to serve a relevant ad to that user. It might also be helpful for the blog platform to suggest related articles to a reader. Article suggestions could align with the age of the user, so that a reader of a certain age would read articles by someone of their same age if the platform found that helped with retention and engagement.

This problem had a number of challenges that we had to address. First, just the size of the data presented issues. The training set had over 400,000 blog entries, and so running any model or data cleaning, or even simply reading in the data, took a lot of memory and time. In order to deal with this, we would take subsets of the overall data to make it more manageable and gain the insights, but then used the entire set when making the final model. Also, the raw data wasn't perfectly cleaned, and so we needed to take some steps in order to sanitize the data for proper analysis, for instance: grouping blog posts by user, to make sure more prolific writers weren't disproportionately affecting the model. The distribution of ages also showed a strong skew to the right, in that the bulk of bloggers were between the ages of 17-30, but there were still some that were as old as 48 years old and as young as 13 years old. In order to account for this, we used a log transformation on the response variable. This helped create a bit more of a normal distribution of ages. We also found that R and Python had separate strengths when approaching this problem, and so we used the different languages where appropriate. For instance, the text mining tools for building our TFIDF was superior in R; however the data cleaning was much faster in Python.

A similar problem would be anything in which groups of text might be used to predict some continuous data, for instance: looking at tweets about a certain stock could be used to predict price fluctuations. If there were certain words used about a given stock symbol that correlated with the price of that stock, we may be able to make predictions about price movements based on a collection of new tweets. We could also use other types of data besides blog posts in order to predict the same response variable, age. For instance, we could look at newspaper articles and the ages of the authors if available to make predictions about the age of the author a new article. The age of the author could be potentially helpful in understanding if that author may have certain biases based on their age, which could help in the interpretation of their articles.