Spam or Ham

A Data Science project in Spam Detection

Belle Peng | 2018-08-08

Project scope

Today spam accounts for 40% of all email traffic; the average person receive 121 emails a day, that means without spam filter, we would be getting over 200 emails a day, and most of it is junk or straight up unpleasant! The goal of this project is to accurately detect which is a spam email versus a real email. The "spam" concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography. In this case, false positives (marking good mail as spam) are very undesirable, therefore my target is maximizing **Precision**.

Data

I found a dataset on Kaggle of 4601 emails with 39.4% spam and 58 features. Given our world statistic of approximately 40% of all email traffic being spam, the 39.4% spam in my dataset sounds reasonable. I do not have an imbalanced dataset problem that I anticiapted. Each observation contains the frequency of 48 words (one word frequency per column). Some examples of these words are "money", "edu", "direct", "free", etc. I also have 6 columns of frequency counts on the following characters: [;, (, [, !, $, #], a column on "capital run length average", "capital run length longest", "capital run ength total", and an indicator whether the email was spam (1) or not (0).

I also have another 2500 actual emails from another corpus, which will require parsing. I parsed these emails to have the same word frequency counts as the first dataset, so I can combine the two datasets. Fortunately I have labels for both datasets.

Project design

I used cross validation to train my models with the target set to Precision. The models I tried are the following: SVM, Naïve Bayes (Binomial and Multinomial), K-Nearest Neighbor, Logistics Regression, Decision Tree, and Random Forest. Each model was tuned using grid search to find the optimizing paramters. Due to my dataset being skewed, I logged all of my features except for the target feature ("spam"). I also scaled the data using standard scaler before feeding into SVM, KNN, and Logistics Regression. My final chosen model was a Random Forest model which reached a 95% precision on teh Training Set and 94% precision on the Test Set. The chosen parameters were using the Gini criterion, max depth of 15, min samples split of 2, and building 350 trees. This model results in 88% recall rate, however my optimizing parameter is precision.

One thing I attempted that didn't work was uploading my data to an AWS SQL database. The upload completed however when I pull it back down, the dataset didn't look right, it created a list intead of a dataframe. Since my dataset was not too large, I did not continue the pursuit down that path.

The primiary data science tool in this project was Python SkLearn. I also used Flask to build a web app to predict whether some text or an uploaded .eml file is a spam for demonstration purposes. To see this web app, please go into the web_app folder on github and start it on command line using "python app.py". A static picture of the web app is also in the PowerPoint deck.

Results

My final model reached a 95% precision rate, and is able to successfully classify whether some input text or a .eml file is a spam or ham. In in the next iteration of this project, I will engineer more features. I will disregard the 4600 pre-processed emails, and just work with the 2500 .eml files so I can engineer more features, such as hyperlinks, more keywords specific to pornography or free-money scams, and do misspelling by matching to a dictionary. I believe more features in my model will improve the results to >99%.