Jacob Stahl

Data Mining

Project 2

Fake Job Posting Classifier

Data set : <https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction>

Source : http://emscad.samos.aegean.gr/

This data set contains 17,880 online job postings flagged as real or fraudulent. 17,014 are real and 866 of them are fraudulent. Each item contains raw text for the company profile, job description, requirements and benefits. There are also binary and nominal attributes that contain information like the job function, industry, and whether or not the job is telecommuting.

During the cleaning phase, many of the attributes had to be removed because they had to many null values. Others had to be removed because they had to many categories and were not standardized, like location and job title. I decided to remove all attributes that were not raw text, and combine all of the raw text for each item into a single string. These strings were then turned into a vocabulary vector that counted the occurrences of each word in the strings.

**Setup Phase**

This data set is very unbalanced because for every fraudulent entry, there are about 19 real ones. To re-balance it, I randomly under-sampled the real entries to match the number of fake ones. After re-balancing, I used 4 fold cross validation.

**Modeling Phase**

I decided to use a multinomial Naive Bayes classifier, and a Decision Tree for my models. Naive Bayes classifiers perform well for detecting spam emails and text messages because they pick up on the probability of certain words being linked to spam, like “won free Nigerian Viagra”. I used a Decision Tree so I had something other than a baseline to gauge the performance of the Naive Bayes classifier. The dummy model I used to measure the baseline performance picks the most common target it sees in the training set, on the test set it scores very close to 50 percent. When training the models, I used an 70 / 30 split for the training set and the test set.

**Evaluation Phase**

code outputs :

cross validation using naive bayes

cross validation average accuracy : 0.8915

cross validation standard deviation : 0.014

training…

Dummy

baseline accuracy : 0.4885

Naive Bayes

model accuracy : 0.9192

confusion matrix (real = 0, fraud = 1):

[[240 26]

[ 16 238]]

metrics :

false positive rate: 0.0977

false negative rate: 0.063

Decision Tree

model accuracy : 0.8173

confusion matrix (real = 0, fraud = 1):

[[209 57]

[ 38 216]]

metrics :

false positive rate: 0.2143

false negative rate: 0.1496

These results were obtained with a Numpy random seed of 1. My results vary slightly with different seeds. At first glace, the text in the fraudulent job postings doesn’t look different than the text in the real ones the way spam emails would. The Naive Bayes classifier is the best model according to every metric, and performs remarkably well at 91.92% accuracy, which is much better than I was expecting. Both models are inclined towards false positives to a similar degree, which would be a good thing if one of them went into production on a real website. If a job posting is incorrectly flagged as fraudulent that flag can always be appealed by the company that posted it, but fraudulent postings that slip by the model could damage the website’s reputation and cost money.