Pizza Learning:

Final Project Report

Ricardo Barrera, Denis Vrdoljak, Max Yan, Linfang Yang

UC Berkeley, School of Information

W207: Machine Learning

Random Acts of Pizza Dataset

13 December 2015

**Problem Definition**

In this project, we use the Reddit dataset, Random Acts of Pizza, where users post requests for a free pizza. Our goal is to predict whether the request will be successful and result in a free pizza.

**Kaggle project link**

<https://www.kaggle.com/c/random-acts-of-pizza>

**Input Data Preparation**

Each request is represented in JSON format, containing attributes such as; request title, request text, comment, requester id, etc. Additional details are available here: <http://cs.stanford.edu/~althoff/raop-dataset/>. There are a total of 5,671 json objects provided in the dataset. We divided the data into a 3969 JSON training dataset and a 1702 JSON test dataset.

**Feature Selection and Creation**

We set up a baseline using one attribute, request\_text\_edit\_aware, which is is the “edit aware” version of "request\_text." Specifically, this is the request\_text field after applying a set of rules to strip out edited comments.

**Data Cleansing and Pre-Processing**

We tokenized the text fields using a simple CountVectorizer, and converted the collection of training data text documents into a matrix of token counts. Then, using the same vocabulary set on a CountVectorizer model for the test set, we generated test token count data.

When processing the data, we tried different text preprocessors. Some of the preprocessing steps included treating data as all lowercase and stripping white spaces. It turned out they did not significantly improve our results. We chose a default CountVectorizer using all lowercase text.

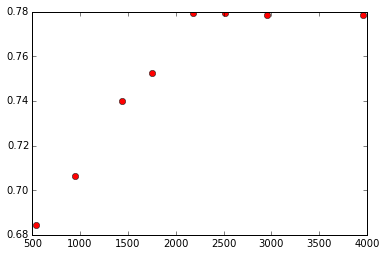
We fit the CountVectorizer tokenized data to a Logistic Regression model, and achieved a baseline F score of 0.67.

**Feature Engineering**

To improve the performance of our model, we looked at a range of features and compared the results on accuracy of including different ones. One example of a feature that resulted in a significant performance gain is requester\_subreddits\_at\_request, which is the list of subreddits in which the author had already posted in at the time of the request. We wanted to see if the requester who received pizza belongs to a particular group. We used BernoulliNB to cross validate.

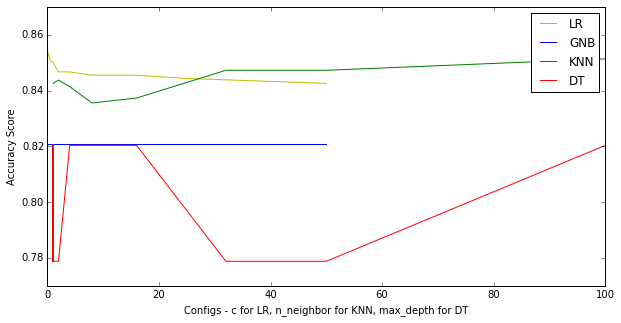
Our investigation shows the data is a good predictor so we included this attribute in our transformed data before we run the final logistic regression.

We also explored with some other attributes such as unix\_timestamp\_of\_request\_utc (Unit timestamp of request in UTC) hour part, request\_title length, and got slightly improvement. An example of our analysis, in the follow chart, shows the effect on accuracy at various vocabulary sizes when adding the tokenized features from the Request Title.



**Conclusion**

Satisfied with our features selection, application of CountVectorizer, and Logistic Regression model, we compared the accuracies of different models with different parameters. The follow graph shows our results.



As a final result, we were able to score a 0.86 accuracy with a Logistic Regression model, a C = 0.001, and our resulting Feature Engineering and selection. This result ranks 13th at the in the overall Kaggle competition. We are considering using our results and findings to submit a Random Act of Pizza request for a Pepperoni Pizza with extra cheese.