Yelps' Review: Individual Rating Prediction

Ching-Han Chiang ***Jianming Zhou***

Project Advisor: Brian Dalessandro

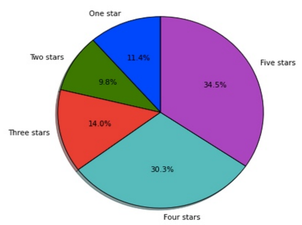
Introduction

Online reviews of local business become the 'social proof' when consumers face too many choices. Yelp is one of the most commonly used sites to search for reviews about local business, and has became an important reference for making consumer decision. In this project, we are interested in predict the outcome of an individual rating on Yelp, given the text content. Various machine learning algorithms are applied on text-related features in the hope of predicting whether the business will have good rating or not. Ideally, the well trained classifier could be used as review-to-rating 'converter' under yelp standard. Specifically, an individual business review content from other review sites, such as Google Places, Yahoo local listing, Angie List and Four square, could be re-rated using the trained classifier under 'Yelp's standard'. Beside the business point of view, our goal in this project is to build the best classifier that beats the baseline and has good prediction accuracy.

Data Preprocessing

➢ **About Dataset:**

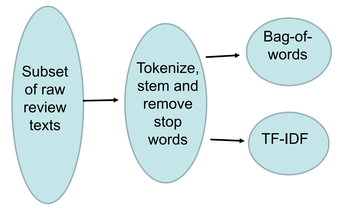
- The dataset is from Yelp Dataset Challenge. Dataset contains 1569264 reviews with information about type, business_id, stars, text, data, and votes. We are only interested in using text to predict the rating stars. To guarantee the quality of reviews, we only use the subset of 557187 reviews that has been voted by other users. And randomly selected a subset of those reviews for training and testing. The target variables 'stars' is in form of number 1 to 5 (1:the lowest through 5: the best). The following figure illustrates the proportion of each star type in the Dataset:



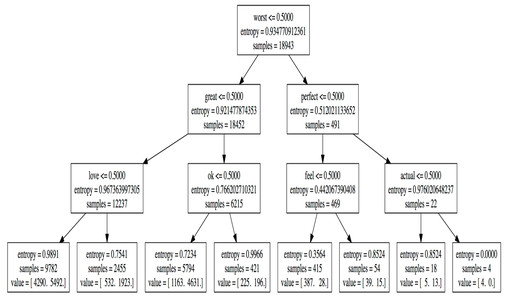
➢ **Feature Selection and engineering:**

- Tokenize each review to generate a word list
- Do word stemming and stopword removing
- Consider the top K sorted features to use for training
- Using bag-of-word as default strategy, also consider tf-idf
- Try PCA for dimensional reduction

➢ Two feature-engineering approaches:



➢ Word visualization one: In order to visualize the some top-most relevant features/words for classification, we use decision tree to display those features based on the information gain-entropy criterion and limit the tree depth to 4 .



➢ Word visualization two: 100 top-most relevant features/words extract from logistics regression:

{**Note**: The size of word/feature is proportional to the value of log(odd-ratio) in the logistics regression.}

Data Analysis and Modeling

➢ Rating reviews are classified into two groups, good rating review(4 or 5 stars) and bad rating review(less than 4 stars). The good reviews contains about 65% of the selected data.

➢ **Baseline model**:
Decision Tree with bag-of-word features is used as the baseline model.

➢ **Models Used for classification**:

- K-Nearest Neighbors with 3-NN
- Random Forest
- Support Vector Machine-RBF Kernel
- Logistics regressions with regularization: L1 and L2
- Adaboost with decision tree classifier as based estimator

➢ **Evaluation Metrics:**
Classification Accuracy(total correct predictions/total number samples) is used for performance evaluation.

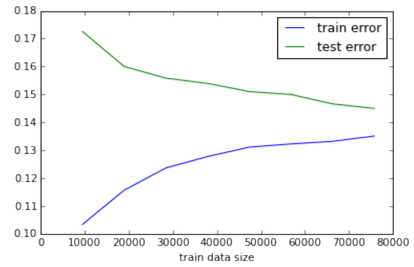
➢ **Models Selection and Performance evaluation:**
We limited the features dimension to 2000, trained each classifiers on the training dataset, adjusted the possible parameters for each model. The most appropriate parameters are selected for each classifiers based on the model performance on training set and validation set.

	Train set accuracy	Validation set accuracy
Logistics	88.86%	83.82%
Support Vector Machine-RBF	86.79%	83.58%
Random Forest	99.98%	83.46%
Decision Tree	77.11%	72.84%
KNN	78.54%	66.42%
Adaboost	99.98%	70.84%

➢ **Notes:** Logistics regression , SVM and Random forest perform better among all other models. However, the kernel based SVM requires lots of computational power when the sample size is big. And it's performance is not better than Logistics and Random Forest, its probably not a good idea to use SVM in this case.

- **Models Performance and Data size**
(training/validation 0.85:0.15):

With fixed model complexity L1 and C=0.15, getting more data for training is likely to reduce the validation error. Based on the figure shows below, it seems like the most appropriate data size for L1 logistic regression is around 7500 with the validation error down to around 14%. And L2 logistics has the similar performance. After the 7500, adding more data doesn't seem to have significant improvement on the model performance.



Conclusions

- Logistics regression , SVM, and Random Forest have performance better than other classifiers. However the run time for kernel based SVM is pretty long and requires lots of computational power.
- Data size of 7500 seem seems like the most appropriate data size to for training L1 or L2 logistics model.
- Future work:
  - In our case, we tried TF-IDF for feature engineering. However, the result we got is similar with Bag-of -word in term of model performance. Someone may try N-grams to see if this help for model performance. Beside predicting review rating, extracting useful information that highly associated with review rating may help business owner to understand what customers need in general.
  - In our study, the best model we can get is logistics regression with accuracy around 86%. If some one could improve the model performance a lot, the classifier could be used to re-rate review content from other website to the class of good or bad review under 'yelp standard/classifier'. If that's the case, reviews for each business from all sites could be re-rated under one standard.