Yelps' Review: Individual Rating Prediction

Ching-Han Chiang Jianming Zhou

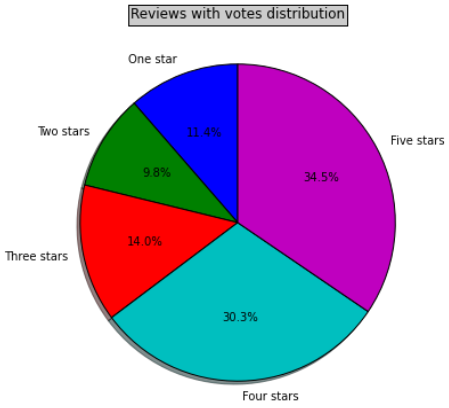
Introduction:

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.

Dataset

- **Data Distribution**

Our dataset is from Yelp Dataset Challenge. There are five separated parts in this dataset. Right now, we are using one part, which is yelp_academic_dataset_review. The yelp_academic_dataset_review contains 1569264 reviews with information about type, business_id, stars, text, data, and votes. To guarantee the quality of reviews, we only use the 557187 reviews that has been voted. Among those features in the reviews, 'text' is the review contents from customers, and the stars in the dataset is the rating star for restaurant, and it's in form of 1-5 (1: the lowest through 5: the best). This following figure illustrates the proportion of each star type in the Dataset.



- **Data Size**

Among 557181 of voted reviews, we randomly select 4% reviews from all, i.e. 22287 reviews for our model training and testing.

- **Two-class mode**

Instead of predicting the number of rating stars directly, we classify our data into two classes based on the number of rating stars, that is, we are interested in predicting whether each individual review is good or bad given text content. The reason for training two-class prediction model is that most yelp users views are not professional reviews. In our opinions,we believe professional reviewers are probably good enough to justify the numbers of stars the business should have, but not the most users/reviewers. For example, when giving the rating stars, most people have difficulty to decide giving 3 stars or 4 stars if the service they got is not that perfect. In this project, we label the 4 or 5 stars reviews as GOOD and the reviews with 3 or less stars as BAD. And the dataset we has contains about 65% of GOOD reviews.

- **Feature**

To generate feature from review context, we first remove stop word from the context and do Porter stemming.Then we use bag-of-word to generate feature vector. Further, we select the most popular 2000 words among all reviews. Then for each bag-of-word feature vector corresponding to a review, we only keep the information of the 2000 words. Hence, each review is corresponding to a dimension-2000 feature vector.

Models Selection

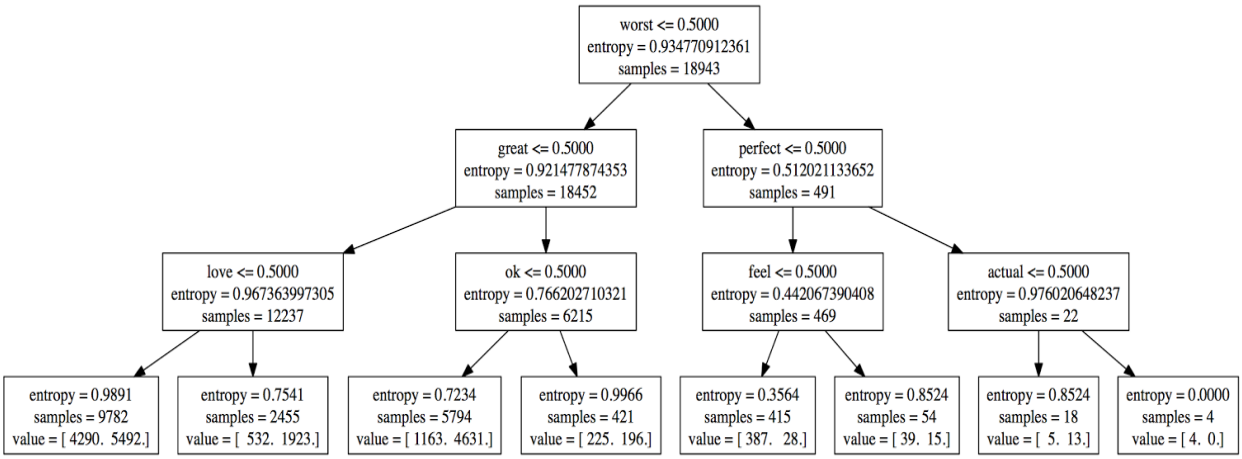
- **Evaluation Metrics**: Classification Accuracy(total correct predictions/total samples) is used for performance evaluation.

- **Decision Tree**

We decide to use decision tree as our baseline model because it's data structure is interpretable and it's running and testing time is linear to the data size and feature length.

- ❖ **Decision Tree - Depth 4**

To have some insight of the data, we first train a depth-4 decision tree and show its data structure as follows. In the data structure, we see that there are many words such as "worst", "great", "perfect" and "love" that are used to describe preference. That is, those words in top level of decision tree are the most relevant for classification. Which seems reasonable as well.



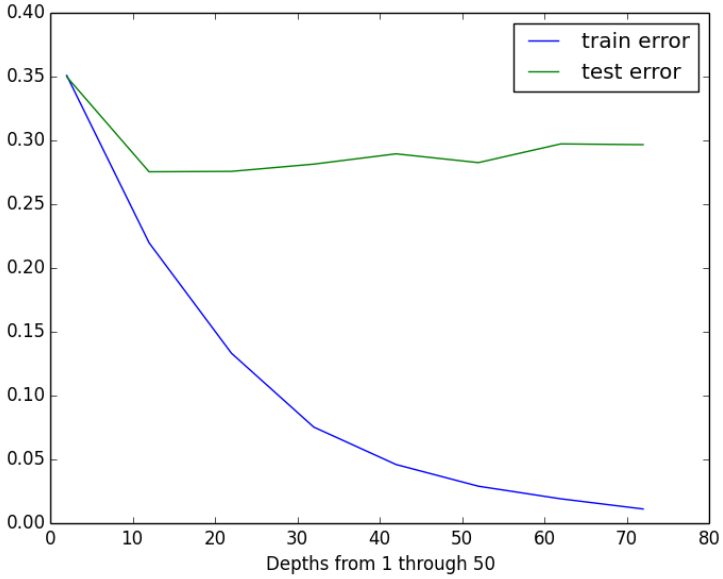
❖ Decision Tree without Depth Limitation

Further, we train our decision tree without depth limitation. The decision tree's test accuracy is 70.84% which is about 6% higher than the GOOD review rate 64.8%. Moreover, we also provide the top 20 features that are used to split the data in the following table where the words with yellow background are used to describe people's preference. Again we observe that 70% of those top 20 features are preference descriptive.

worst	told	excel	perfect	recommend
great	amaz	horribl	awesom	okay
love	bad	best	noth	favorit
delici	rude	terribl	poor	ok

❖ Overfitting of Decision Tree without Depth Limitation

In the previous experiment, our training accuracy is 99.98% which is 29% higher than the test accuracy 70.84%. Obviously, there is overfitting. Hence we train the decision tree with depth ranging from 1 to 71 and then compare training error and testing error. The result is given in the following graph. Based on graph, the testing error achieves the lowest 27.16% at depth 11, then increases with the growth of decision tree. That is, the model complexity of decision tree should be set to depth 11 in this case, the test accuracy achieves up to 72.84%.
We also adjusted the other parameters such as minimum number of samples required to split an internal node and minimum samples leaf, but those adjustments didn't bring significant improvement in testing accuracy.

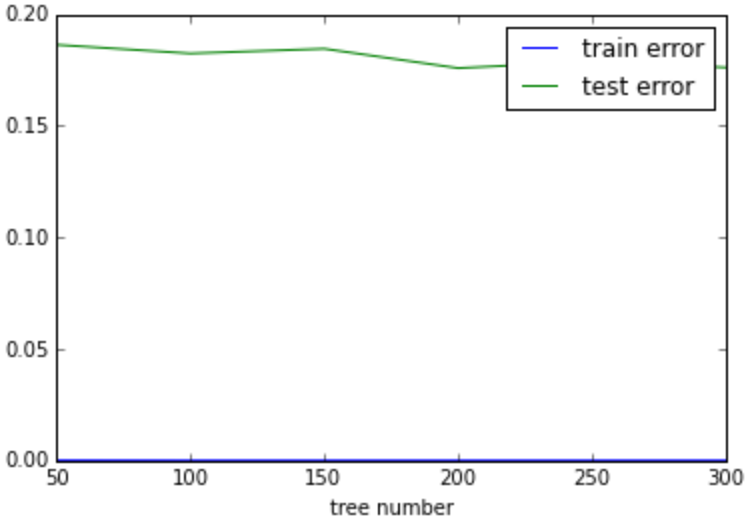


- **Random Forest**

In the training process of random forest, we need to determine two main parameters, number of estimators and maximum depth of each tree.

❖ **Random Forest - Tuning the Number of Estimator**

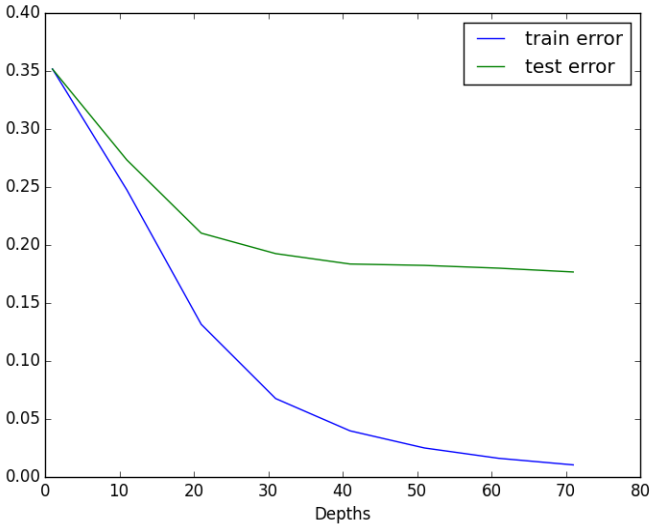
We first experiment the relationship between number of estimators and error rate. The model achieves the lowest error rate 18.75% with 200 estimators.



❖ **Random Forest - Tuning the depth**

Further, we use 200 estimators and then tune the maximum depth from 1 to 71 to get the relationship between training/testing error and depth. The results is shown in the following

graph. According to the graph below, we see that the testing error decreases quickly from depth 1 to depth 21. From depth 21 to depth 71, the testing error slightly decreases and becomes more stable. Since the testing error doesn't increase in trend, we decide not to limit the maximum depth.

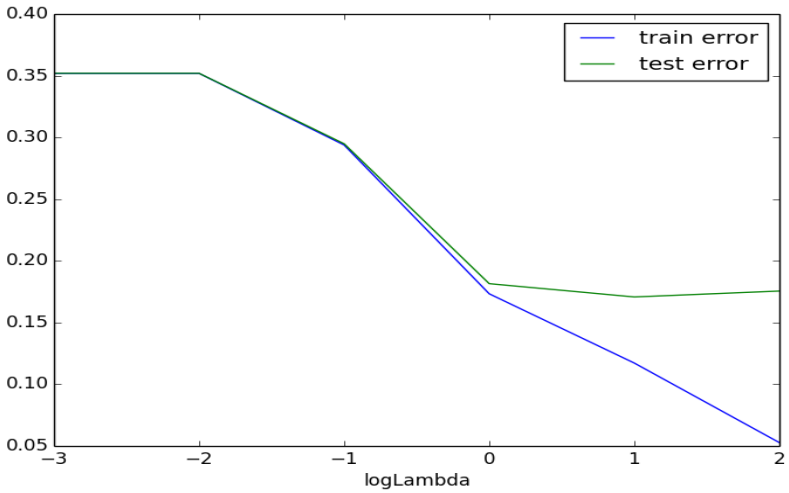


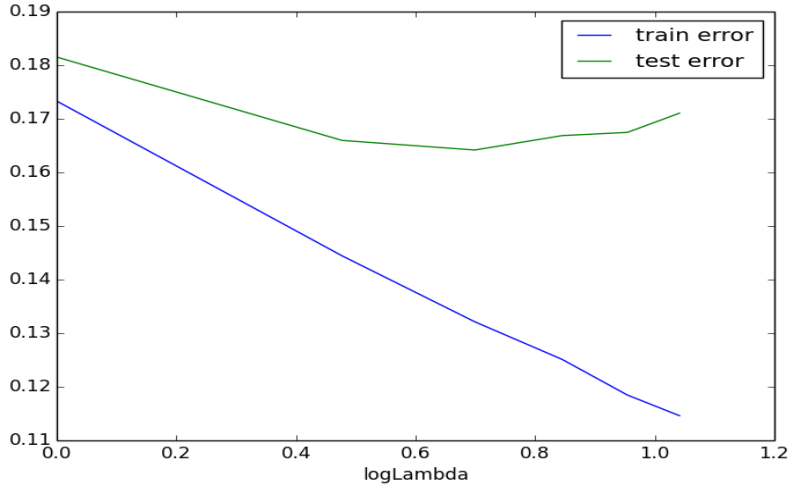
- **Random Forest - Performance**

With optimal model parameters,estimator 200 and unlimited tree depth, the random forest can achieve testing accuracy 83.46%

- **SVM - Regularization**

In SVM with RBF kernel training, we check the testing/training error with different λ values range from 0.001 to 100 as below graph. We observe that its testing error rate achieve the minimum with λ around 10. Equivalently,



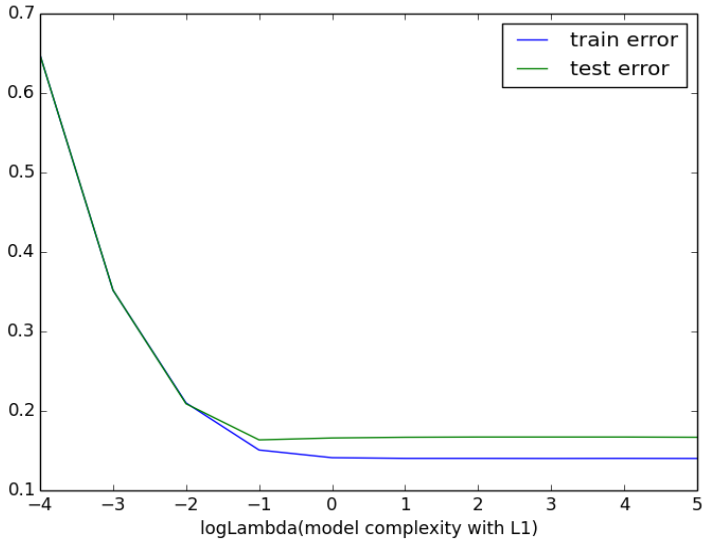


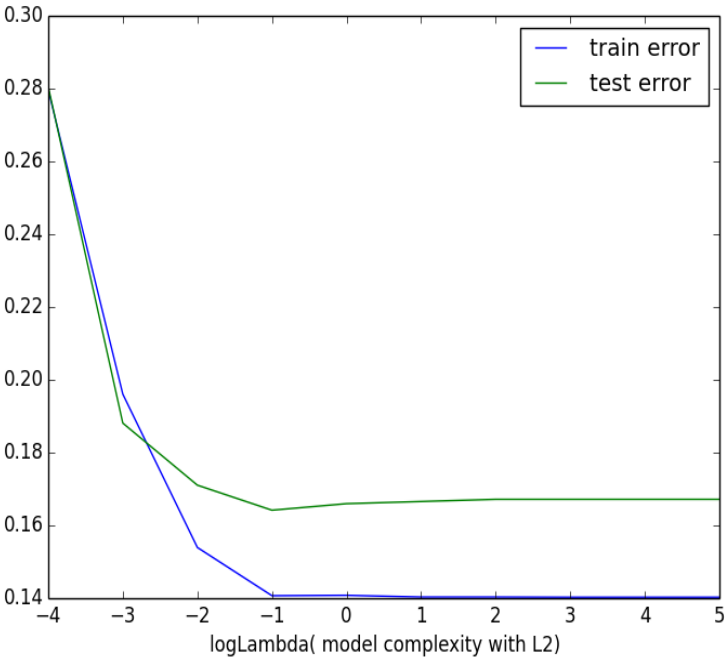
logλ around 1. Then we zoom in λ values(1,3,5,7,9,11) and found that the testing error rate achieve 16.42 % with optimal model parameter λ around 7.

- **Logistic regression**

❖ **Logistic Regression - Regularization**

In logistic regression, we need to decide the types of regularization, L1 or L2. For each type, we also need to decide the λparameter. The following is our results for L1 and L2 regularization. In L1 regularization, the model achieve the lowest testing error rate 16.18% when λ is 0.1. In L2, the model also achieves the lowest error rate %16.27 when λ is 0.1.





From above experiment, we found out that the L1 regression with λ has achieve the lowest error rate 16.18%.

Summary of all models' performance:

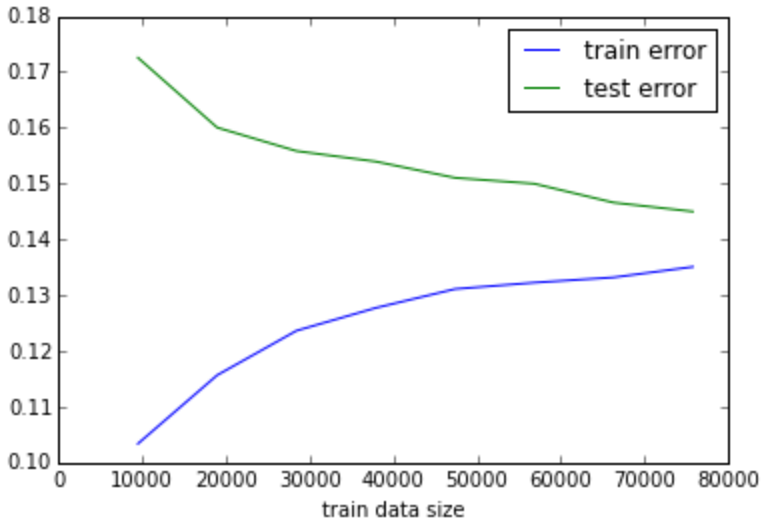
In the following table, all previous model's performance are listed. Among of them, logistic regression achieve the best accuracy 83.82%

	Train accuracy	**Test accuracy**	**Optimal model**

			Parameter
Decision Tree	77.11%	72.84%	Depth: 11
Random Forest	99.98%	83.46%	TreeNum: 200
SVM	86.79%	83.58%	λ: 7
Logistics	88.86%	83.82%	L1 with λ: 0.1

Data Size Analysis

In previous experiments, we train and test those models with 18944 reviews for training (i.e. 85% of 22287). However, we are curious about the relationship between training/testing error and data size. Hence, we analyze the training/testing error of logistic regression with data size ranging from 9472 to 75776 and provide the result in the following.



From above figure, we observe that the gap between training error and testing error shrinks from %7 to 1% when review number increases from 9472 to 75776. That is, with the fixed model complexity L1 and λ: 0.1, getting more data for training is likely to reduce the testing error. Based on the figure shows above, it seems like the most appropriate data size in this case is around 7500 with testing error down to around 14.5%. From this point, feeding more reviews data to the model doesn't seem to have significant improvement on model performance.

TFIDF

About feature engineering, we also try bag-of-word with tfidf. However, its performance is similar to bag-of-word with frequency which is shown as follows.

	Test Accuracy with TFIDF	Test Accuracy w/o TFIDF
Decision Tree	70.22%	72.84%
Random Forest	82.31%	83.46%
Logistics	83.29%	83.82%

Conclusion

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 around 75000 seems like the most appropriate data size to for training L1 or L2 logistics model. 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. For future work, 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 85.5% (after increase data size to around 75000. If someone 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.