Store Survival Prediction on Yelp Datasets

I-Wu Lu, Ying-An Lai, Chun-Yi Yang

Courant Institute of Mathematical Sciences, New York University

Objectives

Modern consumer-end markets are diverse and rapidly shifting. It will be really valuable for owners to know if a business can come through the difficulties in the future beforehand. Given the data from Yelp, we try to information from competitors (stores nearby) and customers (people who left their reviews) to determine whether a store will survive in the competitive market. In this project, a survival prediction model is built to predict if the store will last for half year given data from the first running year.

Data

The dataset is from Yelp Dataset Challenge 2017. There are 114K distinct stores in the dataset, and 4.1M store reviews in total. Business attributes such as *city*, *latitude*, *longitude*, *categories* and *stars* are also included. The total dataset is 4.98 GB.

Preprocessing

1. We first deal with missing values. There are only few fields with missing values, such as *city*. and can be easily inferred from other fields such as *longitude* and *latitude*.
2. We use NLTK, a python natural language Toolkit to process text in reviews. First, we clean up the text and tokenize it into words. Then we part-of-speech tag each word and use a subset of the keywords (labeled adjectives and nouns) that appear most frequently among all reviews to generate feature vectors given a single review.
3. In the end, we adapted normalization for numerical attributes and used one-of-N encoding method for categorical attributes.

Features

Our features include static features and dynamic features. Static features, such as state and name size, are determined once the business started, while dynamic features, such as popularity and stars, are generated using the average among our 1-year observation window.

- state: which state the store located in
- stars: the average stars the store receives
- popularity: how frequently the reviews are received over time
- name size: length of store name
- name polarity: subjectiveness score of store name
- pos. score: sentiment score of positive reviews
- neg. score: sentiment score of negative reviews
- elite count: number of reviews received from elite users
- competitor count: number of competitors within one mile
- key nouns: vector of frequency of the top 100 nouns extracted from shut-down businesses
- key adj: vector of frequency of the top 100 adjectives extracted from shut-down businesses
- review trend: trend of stars from users in the given 1-year data

Challenges

- Imbalanced data
- Data process efficiency

  Some functions of our feature generation stage are time-consuming. Calculating the distance between two stores using longtitude and latitude is one of this example. Given this property and the large size of dataset, we use PySpark library to speed up our feature generation stage.

Modeling Process

We first split our data into training and testing sets by 8:2. After feature generation, we trained various model including logistic regression, linear SVM, and kernelized SVM with RBF kernel. A 5-fold cross validation was applied to get the optimal parameter for these models. Optimized model was selected by Kappa coefficient.

Model Evaluation

Accuracy is not a proper evaluation in our case since our data is imbalanced (open 4: closed 1). Namely, 80% of the stores are still open after the 1-year observation window, if we arbitrarily predict all the results to be 1 (open), we can still get accuracy of 0.8. Therefore, we consider AUC (Area of ROC curve) and **Cohen's kappa coefficient** as our evaluation metrics, which include the information of real T/F ratio. Cohen's kappa coefficient is defined as below:

$$k = \frac{P_0 - P_e}{1 - P_e}$$

where P_0 is the relative observed agreement among true and model prediction, and P_e is the hypothetical probability of chance agreement.

Tools

- geopy: to locate coordinates of stores
- nltk, textblob: to generate word vectors of reviews
- PySpark: to handle massive data feature generation
- Scikit-learn: to train and evaluate model

Acknowledgement

Special thanks to our project advisor Kurt Miller for his insight opinions and class instructor David Rosenberg for his brilliant lecture and efforts on Machine Learning course.

Results

Among all the models, random forest gives the best result to Kappa coefficient. The AUC score shows that our model is better than predict by chance (0.6048 > 0.5).

	Rand-Forest	LinearSVM	LR
AUC	0.6048	0.5918	0.6121
Kappa	0.1956	0.0994	0.1271

Table: Algorithm Comparison

Best setting: *criterion='gini',max_features='log2' min_samples_split=90,n_estimators=50*

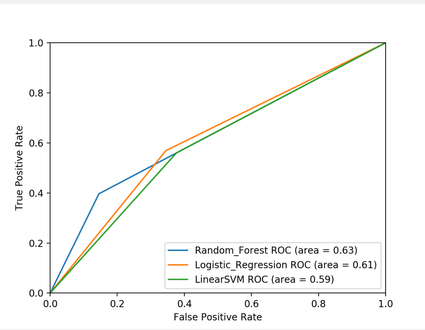

Figure: ROC curve

		Predict Open	Predict Closed
True	Open	17312	1549
	Closed	2029	1036

Table: Confusion Matrix

Future Works

One direction we want to take is to generalize our evaluation using the concept of survival analysis. In survival analysis, we replace current binary target variable with a random variable and try to fit this random variable model. In addition, we can improve our model by taking geographical information into account. In the final report, we will involve a feature indicating how a certain user rates this business in comparison with other competitors nearby.