Kids welcome! — Predicting child friendliness of restaurant & food businesses from Yelp data

Timm Suess

November 12, 2015

Introduction

As parents of young children, eating out can be a challenge. From the availability of food choices to social acceptance of loud children, from the process of ordering to entertainment possibilities — many factors decide if a family meal out in the town will constitute a memorable event or an experience better to be quickly forgotten.

In this research paper, I am outlining a machine learning approach to predict the child-friendliness of restaurant and food venues based on user review data from the popular local business review site Yelp. The research question I am trying to answer is:

> **How well can a "Good for kids" rating of restaurant and food businesses be predicted from business features such as Yelp category, service attributes, city and key words in the business's name?**

I will describe how I constructed the data set from both structured and unstructured data, trained selected the predictive model, and will outline its predictive qualities. A successful predictive model can form the basis of a recommendation engine, enabling parents to discover child-friendly venues without the explicit "good for kids" label. It could serve Yelp to flag potentially fraudulent review entries. It will also allow further research concerning what influences child-friendliness in the restauration industry.

Methods and Data

Data Sources and reproducibility

Yelp.com is a popular crowdsourced review site for local businesses. Yelp's *Academic Dataset* provides a rich collection of information about venues, including a user-driven attribute for child-friendliness. All of the research presented here are based on the data from the *Yelp Dataset Challenge (Round 6)*. All of the code to construct the feature set and predictive model is written in R (V.3.2.2). For space limitation reasons, the code is not shown in this paper, but can be found on my Github repository.

Pre-Processing

Basis of the modelling data is a JSON file called `yelp_academic_dataset_business.json` which contains all of the data fields relating to 60'000 businesses from 10 cities. Through a series of data tidying steps most of the included fields were extracted, namely:

- **Business identification**: Business ID and name
- **Business evaluation information**: Average star rating, number of ratings
- **Business category**: A four-level hierarchical list of categories and sub-categories based on Yelp's business category list.
- **Business attributes**: Features and attributes of businesses such as price range, availability of wi-fi and parking, dietary options, ambience, attire, as well as suitability for activities (breakfast, dancing) and audiences (groups, kids).
- **Location information**: Address, global coordinates

As the outcome variable, the business attribute "`good for kids`" was selected. In about 4% of the listed businesses, this attribute was included twice; in 0.5% of the cases, the two ratings diverged. As no information could be obtained from the Yelp dataset team to explain this inconsistency, I decided to combine the double

ratings into one, discarding the second one if they contradicted each other. The final outcome variable is a Boolean variable called `gfk`, with `gfk = 0` signifying "not child friendly" and `gfk = 1` "child friendly.

The dataset was then reduced to businesses fulfilling two criteria: 1) availability of a positive or negative `gkf` rating, 2) membership in either the "Food" or "Restaurants" Yelp category (incl. subcategories).

In further pre-processing steps, I refined data as follows:

- Using the global coordinates, a k-means clustering was performed to obtain the affiliation to one of the ten urban areas (cities) included in the dataset.
- The businesses name was deconstructed into a stemmed document-term-matrix containing roughly 300 words that appear in more than 0.1% of the business names (excluding English stop words).
- Due to its skewed distribution of the number of business reviews, its log was included as an additional variable
- All of the variables in the dataset were transformed to 0/1 dummy variables, with the exception of the non-binary star rating, and review count/log.

The final dataset featured **20'091 observations of 501 variables**. An exploratory analysis of the outcome variables showed that with 82%, the "good for kids" tags is skewed towards positive ratings, but still above the threshold of 15% to be considered a "rare event":

Table 1: Distribution of outcome variable

	Not good for kids	Good for kids	Total
n	3667	16424	20091
% Total	18.3%	81.7%	100%

Note that this distribution should not be interpreted as "Most food/restaurant businesses are child-friendly": It is far more plausible that there is a selection effect or bias, i.e. that Yelp reviewers are just more likely to write a review or attach a "good for kids" tag to a business if their experience has been positive.

Modelling

To train and select models, I divided the data into a training (60%), a validation (20%) and a testing set (20%). The training set was used to train three classifier models, the validation set to evaluate the models and combine them into an ensemble. The testing set to evaluate the ensemble model.

All missing values (NAs) in the training dataset were imputed by the variable's median. During validation and testing, the imputation rules were based on the medians of the training set.

The data was used for trained with four supervised learning algorithms:

- **Random Forest** (package `rf`): Known as one of the most robust supervised learning algorithms, Random Forests take a boosted approach to decision tree learning. While regular decision trees lead to overfitting, Random Forests construct a multitude of decision trees and avarage them out, increasing the performance of the model. The algorithm was employed using default parameters, except for `nodesize` set to 120 for efficiency reasons.
- **LogitBoost** (package `logitboost`): The LogitBoost algorithm applies boosting to logistic regression modelling by running multiple linear logit regression models (i.e. "weak" prediction models) and constructing a generalized model. The algorithm was employed using default parameters.
- **Neural net** (package `nnet`): Neural networks simulate brain cells and their learning mechanisms, which can be used for pattern recognition and classification. The model I used contained a single hidden layer of neurons. The algorithm was employed using default parameters.
- **General Additive Model** (package `gam`): The general additive model (GAM) is a simple ensemble learning algorithm which combines multiple machine learning algorithms in a linear logit model with the goal to produce better results than the individual models. This alogrithm was applied to the other three models (Random Forest, LogitBoost, Neural Net).

As a selection metric, I chose the optimal in-sample Receiver Operator Characteristics (ROC) value. The ROC, typically shown as a curve, is a standard metric for evaluating machine learning algorithms. It compares the specificity and sensitivity of a model against each other at varying logistic threshold levels.

Exploratory analysis of employed training approaches

The four algorithms yielded different degrees of in-sample ROC values:

Model	In-sample ROC
Random Forest	0.9055043
LogitBoost	0.8423783
Neural Net	0.8203946
General Additive Model	0.8101904

The table shows that, surprisingly, the additive model did perform better than the individual models.

Testing and final model selection

The individual models were tested against the validation set using. As a final step, I tested the GAM model against the testing set and compared its results to the individual models. The final model was chosen among the four options (Random Forest, LogitBoost, Neural Net and GAM) and tested again against the combined testing + validation set. As performance parameters, I have chosen six metrics:

- Accuracy: The ratio of true predictions among all predictions.
- Sensitivity: The ratio of true negative predictions among negative predictions.
- Specificity: The ratio of true positive predictions among positive predictions.
- Kappa: A relative measure of observed accuracy against expected accuracy.
- Balanced Accuracy: The average of specificity and sensitivity.
- AUROC: The area under the ROC curve, a measure for how well an algorithm performs under possible logistic threshold levels.

Results

The performance parameters for the four algorithms are as follows:

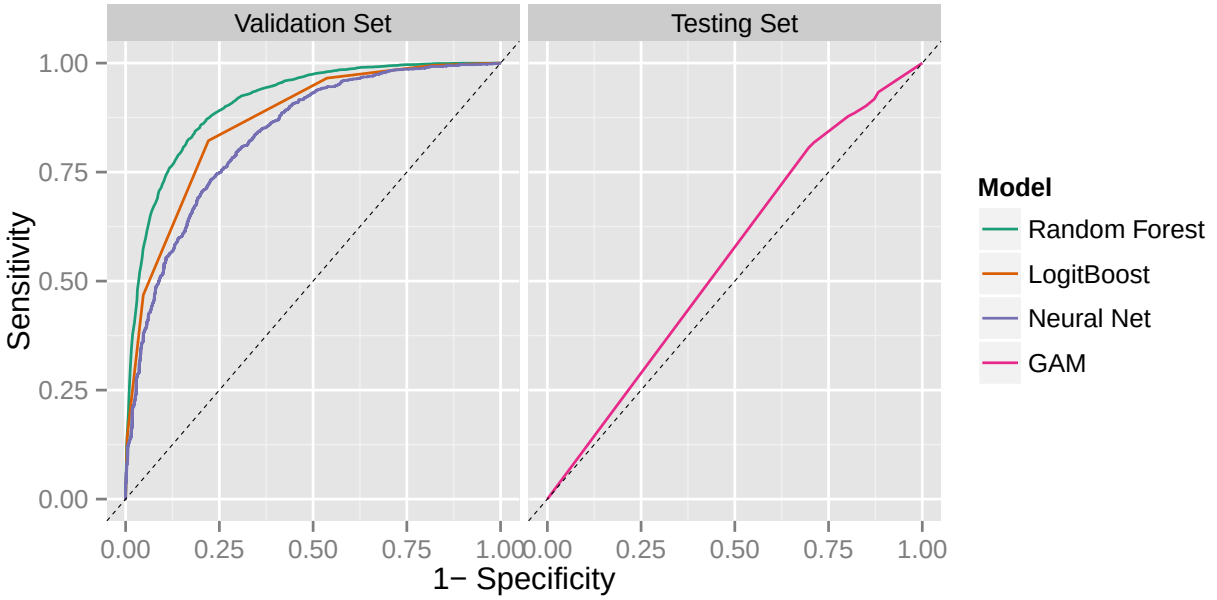
Table 3: Classifier comparison

	Accuracy	Sensitivity	Specificity	Kappa	Balanced_Accuracy	AUROC
Random Forest	0.887	0.966	0.538	0.574	0.752	0.908
LogitBoost	0.873	0.965	0.463	0.503	0.714	0.865
Neural Net	0.844	0.916	0.529	0.463	0.722	0.834
GAM	0.761	0.901	0.151	0.062	0.526	0.555

The below graph shows the ROC curves of the three individual models (based on the validation set) and of the GAM ensemble model (based on the testing data set).

The metrics and ROC curves clearly show that the Random Forest model is the best performing of all four models. Its AUROC value of over 90% is very good; its Kappa value of 57% is fair to good. Due to the positively skewed outcome variable, an accuracy tradeoff on the specificity side (with 54% only slightly above random guess) is present, while the specificity is at an excellent 97%, leading to a moderate balanced accuracy of 75%. In other words: The Random Forest model is very good at identifying child-friendly venues, but performs poorly at identifying child-unfriendly ones.

My (probably naïve) expectation that the GAM ensemble would optimize the joined algorithms to perform better than any individual ones was disappointed — on the contrary: The GAM model performed worse than any of the individual ones.



As I had made no modifications to the original Random Forest model, I was able to test it against a combined validating + testing set without the danger of overfitting. This larger test set showed comparable metrics (Kappa = 57%, Sensitivity = 97%, Specificity = 52%, Balanced Accuracy = 74%).

Discussion

The goal of this research was to find out how well a "Good for kids" rating can be predicted from Yelp business features. The answer is: Quite well, but far from perfectly. While the model is excellent at predicting child-friendly venues, it performs poorly at predicting child-unfriendly ones.

Using the Random Forest model, it should be possible to build a solid recommendation engine. However, there is a not negligeable danger of incorrectly recommending a child-unfriendly venue to parents (a false positive recommendation). The fact that in the data set are 5 times more child-friendly tags than child-unfriendly does not alleviate this danger — as mentioned above, this is likely due to selection effects or biases.

Interpreting the Random Forest model

While the focus of this research is on *selecting* an algorithm and an exhaustive interpretation of the final model would challenge the 5-page limit, I am including below the 25 most important variables from the Random Forest model. Please keep in mind that Random Forest is not the same as a regression algorithm, and that some of these variables are likely inter-correlated.

The most important predictors for a "good for kids" rating (based on the mean decreased Gini impurity) are as follows:

```
##  [1] "attr_alcohol_full_bar"    "review_count_log"        "review_count"
##  [4] "attr_take_out"            "pricerange_3"            "attr_attire_dressy"
##  [7] "stars"                    "goodfor_latenight"       "attr_attire_casual"
## [10] "attr_smoking_outdoor"     "attr_smoking_yes"        "ambience_casual"
## [13] "pricerange_1"             "attr_alcohol_none"       "attr_good_for_groups"
## [16] "goodfor_lunch"            "cat_american_new"        "pricerange_2"
## [19] "attr_takes_reservations" "ambience_trendy"         "attr_caters"
## [22] "term_bar"                 "goodfor_dinner"          "parking_street"
## [25] "attr_outdoor_seating"
```

What we can see is that the important factors relate to the venue's evaluation (stars, review count), pricing, ambience and attire, attributes relating to smoking and alcohol as well as options for when and where to eat (outdoor seating, catering, good for lunch/dinner/late night). The only restaurant category mentioned in the list

is “American (new)”. By comparing the distribution of the predictor against the outcome variable, it is possible to get a (statistically unconfirmed) indication for the direction of the variable.

An interpretation of the variables in Table 3 is as follows: If you want to enjoy a family outing, pick a place that offers take-out or catering, is inexpensive (Yelp price range “1”), has a casual dress code, is non-smoking, known for great lunches and frequented by groups.

If you are out with children, stay away from bars and other places that serve alcohol, venues known for great dinner and late night entertainment, “New American”-style cuisine, expensive, trendy and dressy venues and places that offer street parking and outdoor seating.

With the exception of the last two variables (street parking and outdoor seating), these recomendations seem sensible and hold little surprise value.

Predictor	Direction
attr_alcohol_full_bar	- negative
attr_take_out	+ positive
pricerange_3	- negative
attr_attire_dressy	- negative
goodfor_latenight	- negative
attr_attire_casual	+ positive
attr_smoking_outdoor	- negative
attr_smoking_yes	- negative
ambience_casual	+ positive
pricerange_1	+ positive
attr_alcohol_none	+ positive
attr_good_for_groups	+ positive
goodfor_lunch	+ positive
cat_american_new	- negative
pricerange_2	- negative
attr_takes_reservations	- negative
ambience_trendy	- negative
attr_caters	+ positive
term_bar	- negative
goodfor_dinner	- negative
parking_street	- negative
attr_outdoor_seating	- negative

Outlook and recommendations

There are a number of options to improve the model. On the data side, obtaining more data on child-unfriendly rated businesses would be helpful to deal with the unbalanced outcome variable. One source could be the unstructured review texts, which might contain comments on child-friendliness. On the model side, the imputation process is the biggest weakness in my model and could be much improved by employing more sophisticated algorithms of replacing missing values (k-nearest neighbour, mutliple imputations). Expert knowledge could be employed to construct additional or hidden features. Opening times and geographical information from external data sources could be included in the feature set. Finally, additional models such as Support Vector Machines and deeper neural nets could be explored.

As a starting point for proof of concept of a recommendation engine however, the algorithm holds some promise, as well as the chance for additional exploration of variable influence on child-friendliness. As usual, further research is needed.