Learning From Yelp

David Nichols

Abstract

Yelp serves as an invaluable source of information for business owners. This project aims to use machine learning methods to extract information from Yelp and turn that information into usable insights for business owners.

1 Introduction

Yelp is the leading rating and review site for businesses in the United States. For many consumers, it is an integral factor in deciding where to eat, which doctors to see, and where to shop. This has been a growing phenomenon over the past several years, and studies show that the reputation of a business on Yelp can dramatically affect its business performance. A study by Harvard Business School has found that a one star difference in Yelp rating can impact revenue by 5-9 percent. Another study by Berkeley found that a half star increase in rating causes a 19 percentage point increase in the likelihood that a restaurant is fully reserved during peak meal hours. Because of these economic effects, it is critical for both existing business owners, as well as potential business owners, to understand what can be done to garner higher ratings on Yelp.

Yelp generates massive amounts of user data. Most widely known are the reviews and ratings, but there are also tips, which consist of mini-reviews and pointers, check-in data from when a customer decides to check-in to the business on Yelp, and photos. In addition, there is a fairly detailed set of basic business information such as the hours, price range, parking situation, the ambiance, etc.

For this project, the primary input is the text reviews. These are accompanied by location data, restaurant category, parking, and other business attributes. The output is the rating given to the restaurant by the reviewer. The data did require some pre-processing to be used in the desired fashion, which will be discussed in later sections. By taking these inputs to predict a rating, the objective becomes to fit the most accurate model possible, and then understand what inputs/features drive the accuracy of the resulting models. This process reveals the relationship between the inputs and the ratings, and thus useful insight can be inferred for those who depend on good Yelp ratings to make their living.

2 Related Work

Yelp runs dataset challenges periodically, offering rewards to the best research papers that are based on their datasets. Julian McAuley and Jure Leskovec made a submission that uses the review text and ratings to develop a recommendation system using the Yelp data and compares it with similar datasets from Amazon and others. Huang et. al pursue a somewhat related topic modeling approach using Latent dirichlet allocation (LDA). Both of these works delve much deeper into topic modeling of the text, and their findings would be useful in continued work on this project when it comes to extracting deeper insight from the reviews.

3 Dataset and Features

3.1 Dataset

Yelp publishes an openly available dataset for academic research purposes. This makes getting access to the full richness of their data much easier. The data they offer certainly does not give full coverage, but they give near full coverage of several metropolitan areas. In the current iteration, Pittsburgh, Charlotte, Phoenix, Las Vegas, and Montreal are included among others. I have limited the scope to cover these 5 areas, as they were the ones with the largest quantity of examples, and gave a reasonably broad diversity of coverage, which would help in the ability to generalize from the findings of the project.

The dataset includes all types of businesses - restaurants, parks, doctors, and many others. For the purposes of this project, I limited the scope to only cover restaurants. Besides being by far the most common category in the data, they still represent the most quintessential use of Yelp on a daily basis. After filtering for location and the restaurants category, the dataset consists of:

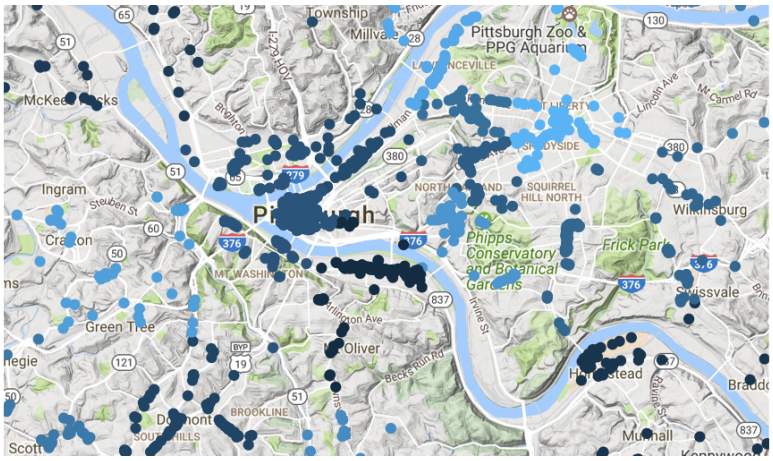
- 22,816 Restaurants, each with a set of basic attributes
- 1.54 million text reviews, each with a 1-5 star rating

3.2 Features

To get the text reviews into a more meaningful form, as a pre-processing step, I used natural language processing techniques to assign a sentiment score to each review. Each of the reviews is typically about a paragraph in length. The first step in this was breaking each review into its component words, using what's know as bag of words. There are typically two versions of bag of words - one that includes word counts, and one that simply tracks word presence in a document. I chose to use word counts to pick up any additional expressiveness being conveyed, as some reviews tend to be quite long and can have repeated words, thus word presence features can miss the ratio of positive/negative words in sentiment analysis in such cases.

After breaking the reviews into word count features, I then applied two sentiment metrics to the word counts. For one, I used the training set to find the mean review rating that includes the term. This was intended to give a Yelp-specific usage weighting, as it was being learned directly in this context. The other metric used was AFINN. This is a metric developed Finn Årup Nielsen in the microblogging space (ie - Twitter). It contains 2,500 words and assigns a value of -5 to 5 to each word to convey the valence of the term. My approach was to convert the AFINN score into the 1 to 5 point scale of Yelp ratings, and then average the two metrics (average Yelp rating with word and AFINN score of word). The intuition for using a Yelp-specific and a non-Yelp sentiment measure was in effect to prevent some degree of overfitting to the words in the training set. This combined metric is what was applied to word counts to establish the sentiment score feature for the project.

The other major feature engineering task for the project was turning the restaurants' lat/long location data into a meaningful metric. Instead of looking at pure location data, a localized clustering (via K-means clustering) was performed to estimate the density of the area where a restaurant is located. The k-value (number of clusters) was set independently for each metro area in proportion to the total quantity of restaurants in the metro area. The cluster assignments and resulting within cluster dissimilarity were normalized and used as a metric to measure how dense of an area the restaurant was located in. A visual of the clustering method applied to the Pittsburgh area can be seen below:



4 Methods

In terms of learning algorithms, the following supervised learning algorithms were used:

4.1 Linear Regression

Linear regression simply tries to fit a line (in many dimensions in this case) to data and gives a continuous response. Its hypothesis class is represented by $h(x) = \theta^T x$, a simple linear combination of the weights and features. The corresponding cost function is $J(\theta) = \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y})$. The optimization problem of minimizing this cost can be solved in closed form via the normal equations, or by an iterative method such as gradient descent. The output of this algorithm is the vector θ of learned weights for the features.

4.2 Ridge Regression

This is a variant of linear regression that adds a penalization term to the weight vector θ. After adding this penalty, the cost function becomes $J(\theta) = \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) + \lambda||\theta||_2^2$, where λ is a tuning parameter that controls the penalty for the growth of the θ vector. This restriction is used in an effort to control the overfitting of the model, with the purpose of helping the model generalize to unseen data better by not trying too hard to fit every bit of variance in the seen data.

4.3 Logistic Regression

Logistic regression is a common classification model defined by the hypothesis class $h(x) = 1/(1 + e^{-\theta^T x})$, also known as the logistic or sigmoid function. This function returns values in the 0 to 1 range, so is naturally well-suited to binary classification problems. It is fit using maximum likelihood which starts with the likelihood $L(\theta) = \prod_{i=1}^{m} h_\theta(x^{(i)})^{y^{(i)}}(1 - h_\theta(x^{(i)}))^{1-y^{(i)}}$, takes the log of the likelihood, and solves for the derivative of the log likelihood with respect to θ to find the likelihood-maximizing value of the weight vector θ.

4.4 Gaussian Discriminant Analysis

Gaussian Discriminant Analysis is another classification algorithm, but unlike the previous algorithms, is a generative learning algorithm, meaning that instead of solving for $p(y|x)$ directly, it solves for $p(x|y)$ and then infers $p(y|x)$ using Bayes' Theorem. The assumption made is that $p(x|y)$ is distributed according to a Gaussian (normal) distribution. The assumptions we use are thus:

$$p(x|y = 0) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x - \mu_0)^T\Sigma^{-1}(x - \mu_0))$$
$$p(x|y = 1) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x - \mu_1)^T\Sigma^{-1}(x - \mu_1))$$

$p(y) = \phi^y(1-\phi)^{1-y}$, where ϕ is the probability of $y = 1$

Similarly to logistic regression, this model is fit by taking the derivative of the log likelihood of the data and solving for the parameters.

4.5 Support Vector Machine

There are a few different interpretations of support vector machines, but the most intuitive is to think of the SVM as equivalent to choosing a hinge loss function, which is given by $J(y|x;\theta) = [0, 1-\theta^T x]_+ = max\{0, 1-\theta^T x\}$. Using this interpretation, we can fit the data using an optimization algorithm such as stochastic gradient descent. It should be noted that the hinge loss has a zero gradient any time the margin (ie - $\theta^T x$) is greater than 1. It fits to the points near the thresholds of the classes, and treats all non-'support vectors' the same. After learning θ, classifications are made using $sign(\theta^T x)$.

5 Experiments/Results/Discussion

5.1 Model Accuracy Results

The purpose of this learning exercise is to extract insight into what features drive high Yelp ratings. As the goal is insight, emphasis was purposely placed on choosing parametric models that learn weights for each feature so we can infer levels of importance. After training several models, here are the prediction accuracy results:

Model	Training Accuracy	Test Accuracy
Linear Regression	0.665	0.648
Ridge Regression	0.659	0.651
Logistic Regression	0.658	0.649
Gaussian Discriminant Analysis	0.685	0.678
Support Vector Machine	0.671	0.661

5.2 Feature Importance / Inference

One way of evaluating feature importance is by looking at the results of stepwise feature selection. This looks at which features the algorithm would choose if it was limited to only being able to choose a subset of the features. Below are the results of forward stepwise selection (note - forward selection is a greedy algorithm so it is not guaranteed to find the global best subset of size x, but it does give a good general sense of variable importance):

Choice	Feature
1	Review Sentiment
2	Category = Fast Food
3	Review Count
4	Latitude
5	Category = Burgers
6	Category = Chinese
7	Category = Pizza

The results here are mostly intuitive, with a few surprises. Fortunately, review sentiment as measured by the pre-processing step from earlier is turning out to be an important predictor of rating, which is intuitive - positive and negative words showing up in reviews should give us a decent sense of how the person will rate the restaurant. A handful of restaurant categories are important predictors of rating. All of the categories here (Fast Food, Burgers, Chinese, Pizza) were assigned negative weights by the models, which, despite being some tasty choices, doesn't come as a huge surprise. Review count being a strong (positive) predictor is also intuitive - good restaurants tend to become popular and

generate more traffic, and thus more reviews. The unfortunate part is that this isn't a particularly actionable insight (other than following the other insights to establish a good restaurant). The inclusion of latitude, but not the derived location density feature from the clustering pre-processing was certainly disappointing, and would require more digging to find intuition for. In the following sections, a couple of these predictors are explored more in-depth.

5.3 Sentiment Analysis

The below terms inferred the most sentiment in each direction:

Most positive terms	Most negative terms
superb	furious
outstanding	pissed
amazing	worst
fantastic	liar
heavenly	disgusted

There are no major surprises here. In terms of messaging to restaurant owners, the main theme in the positive terms was food quality. On the negative side of the ledger, the first two terms 'furious' and 'pissed', and the fourth term 'liar' actually have little to do with food quality in usage context, and are more related to customers having bad experiences at the restaurant. The takeaway here is if you want to avoid very negative ratings, keep a focus on customer service. The terms 'worst' and 'disgusted' are more food quality-related, so it is also imperative for restaurants not to serve up sub-par food.

5.4 Restaurant Categories

One of the most interesting experiments was fitting models using only restaurant categories as features. The category features whose weights had the largest absolute values are presented below. This information may help those planning to enter the restaurant business to choose a category that is well-received, and may thus have a higher potential for success:

Highly rated categories	Poorly rated categories
Delis	Chicken Wings
Polish	Fast Food
Vegan	Buffets
French	Tex-Mex
Peruvian	Burgers

6 Conclusion/Future Work

6.1 Conclusion

The various models used in this project had relatively similar performance accuracy on the dataset. With that consistency, the focus was on inference from the model fits. From the sentiment analysis, it was highlighted that restaurants should focus on making great food and avoiding negative incidents on the customer service side. It also appears that being in a popular location doesn't necessarily translate into higher ratings. It's also important for potential restaurant owners to carefully consider the category they choose to enter, as that will certainly influence the ratings they can expect.

6.2 Future Work

In continuation of this project, the use of more sophisticated natural language processing methods, possibly using deep learning, on the reviews would be very helpful in extracting features from the language data. Also, looking into the user network could help understand different types of customers and allow a deeper understanding of what they value in a restaurant. Further, there are accompanying business photos in the dataset that could be processed using computer vision techniques and used to generate additional features for the models.

References

[1] Michael Luca. *Reviews, Reputation, and Revenue: The Case of Yelp.com.* Harvard Business School, 2011.

[2] Michael Anderson & Jeremy Magruder *Learning from the Crowd: Regression Discontinuity Estimates of the Effects of an Online Review Database.* The Economic Journal, 2011.

[3] Julian McAuley & Jure Leskovec. *Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text.* , 2013.

[4] James Huang, Stephanie Rogers, Eunkwang Joo. *Improving Restaurants by Extracting Subtopics from Yelp Reviews.* , 2014.

[5] Finn Arum Nielsen: AFINN
`http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010`

[6] Wikipedia: Yelp
`https://en.wikipedia.org/wiki/Yelp`