Yelp Mining Progress Report

Tyler Lugger Amogh Jahagirdar Elias Bezanis

1. PROBLEM STATEMENT

In this project, our groups motivation is to use the data collected by the review service Yelp to help restaurants that are being reviewed. Yelp has been around for a long time and has an enormous number of reviews for restaurants all across the globe. These can be created by anyone with access to the website and can help or hurt a restaurant. It has been thought that Yelp reviewers can sometimes hold restaurants hostage by holding the power to influence how others perceive a restaurant. Good reviews can help the popularity of restaurants by bringing in more people that want to try it for themselves.

We want to use Yelp data to give more power back to the restaurants being reviewed. By mining data about the users that frequently use the review service, we hope to create a predictive model of user behavior. This would allow us to find out several aspects about Yelp users and pass that information on to the restaurants themselves.

2. LITERATURE SURVEY

Since the Yelp Data set challenge, is a public challenge all across the world, there are many existing works. Furthermore, the competition is on its 9th round, so there is a plethora of existing work from the previous rounds. An analysis, focusing primarily on the winners of the previous rounds, their questions, as well as their approaches would certainly help guide the direction of this research.

One group of winners in the first round of the competition utilized the Latent Dirichlet Allocation algorithm in order to extract latent subtopics from Yelp restaurant reviews[1]. The LDA model is one which allows a set of observations to be described by "hidden" groups, where these groups explain why there is some similarity in the data. In this group's case, it would allows them to specifically isolate which features customers care about when they are about to rate a restaurant. In terms of pre-processing, this group isolated businesses which were only restaurants, and reviews only targeted at restaurants. Since LDA, relies on the concepts of topics, the text of a review had to be assigned into different subtopics. Essentially, this splitting of text and assigning into various categories is a form of Natural Language Processing. This group also used tools such as Python and other standard visualization/numerical analysis libraries. After, their analysis, they discovered that the keyword "service" in the review [1], was essentially the main factor for a given rating.

A group of winners in the fifth round describe a multi-instance classification technique that goes beyond classifying at the group level[2]. In other words, the primary focus for this group, is not necessarily the mining results, but the generation of a new technique that does more than simply take a given object and classify it into a group (for example positive or negative); it can use these group level classifiers to train what they call "instance-level classifiers"[2]. These instance level classifiers can extract specifics from a given object and then perfect the model. For example, a positive review can still have negative comments. Instance level classifiers would use the group classifier to first put the positive review in the right category, but then extract the negative comments, and further perfect the existing model(See [2] page 1). This can be particularly useful, to perfect a business. Even though many may find that a given business has high quality, analyzing keywords that point out small errors, can help this business elevate their customers' experience. They describe their model, by using cost functions and other relatively complex mathematical expressions. For pre-processing, they had a combination of the Yelp data set, as well as data from Amazon reviews. Then they performed natural language processing, in combination with a group classifier, to extract words into a basket of categories (including "Very Positive", "Positive", "Neutral", "Negative", and "Very Negative") (See [2] page 6). After rigorous analysis, they concluded that their technique was not only accurate on the real world data sets, but also were relatively scalable.

A group of winners in the seventh round also won the grand price for developing a new technique, which they called "Semantic Scan"[3]. Essentially, the goal of semantic scan is to provide quick detection and characterization of "emerging topics in text streams" (see [3] page 1). They argue that existing similar techniques have shortcomings for rapid detection of these emerging topics. They specifically mention that LDA, Latent Dirichlet Allocation is too slow for

analyzing "web scale text streams" (see [3] page 1). Primarily, semantic scan focuses on utilizing a "contrastive LDA model with spatial scan statistics" (see [3] page 3), where topic modelling and assignment is fundamentally changed from different models, with a focus on rapid analysis as well as analysis with noisy text. To pre-process, they setup the Yelp data in such a way that the data simulates an on going business event, as their technique is focused on analysis of a text stream. For example, they simulate a surge of business for a given restaurant and set up a text stream of the reviews, and apply Semantic Scan on it, to see how rapidly and accurately it can assess the noisy text data(See [3] page 8). At the end, they concluded that their model was also effective, and provided statistical tests to further illustrate this.

It's important to note that there were many more winners, and general participants in this competition [4]. Analysis of even more participants, would further aid in not only perfecting the research/evaluation process, but in gaining an understanding of why certain models work in specific cases.

3. PROPOSED WORK

Our team has developed a variety of strategies to process the Yelp data set in an innovative and efficient way. With the help of MongoDB we can begin processing the JSON formatted data without setting up that much of our own infrastructure. In our pre-processing phase we plan to use many different methods for a variety of effects. We are first going to reduce our data. The Yelp data set isn't massive, but for our purposes certain subsets such as the picture subset aren't necessary for predicting user behavior. Then we will clean our data for restaurants only. Based on the restaurants we find, we are discussing integration of a weather data set to cross-reference reviews and types of weather, because we believe there may be a correlation here. Finally, we need to transform the data such as text reviews or check-ins into a more manipulable and standard type of result. To transform raw text into manipulable data we have a program called "Heavy Moose"[5] developed by a group member that can be reconfigured to parse text strings and count important values.

In terms of an end result, our goal is different than those of prior competitors. From the data we process we not only want to predict user behavior, but actively incentivize consumers and producers based on our predictions. This can be done in a variety of ways such as couponing before a customer goes to a restaurant, or letting a restaurant know that they may have a spike in customers today so they should get more staff. The whole purpose of Yelp is to make the service industry more effective. By bringing the consumer and producer closer together through predictive modelling, a type of communication that will be beneficial for both parties can take place. To evaluate the effectiveness of our result, we would need a few months to look at the "newer" version of the Yelp data set and verify our predictions.

4. DATA SET

Our data set comes from the Yelp data challenge. Yelp has selected a portion of their review data that anyone can use for academic purposes. This data set which is nearly five gigabytes in size, contains millions of tuples spread across five JSON files. These files contain all of the information we will need about Yelp users, reviews on businesses, and information on the businesses themselves.

Since this data set is part of a challenge, our team will have the opportunity to submit our data mining results to the challenge to win one of the many prizes that yelp gives to teams that work on their data set. If we are able to find anything useful from this data set, we have the potential to create real change in Yelp and how businesses use Yelp to maximize customer satisfaction.

Our weather data set was requested from the NOAA(Nation Oceanic and Atmospheric Administration). They maintain a database of weather from stations around the world (unfortunately not Canada, the UK, or Germany because there are cities from these countries in the Yelp data set). The data can be downloaded as a CSV or PDF. The original data set requested is hourly data for 8 US cities for 10 years 2007-2017. The data set is 167megabytes in CSV format, contains 870,00 rows, and 26 columns. We do not plan on using every year and every column.

5. EVALUATION METHODS

To evaluate the results of our data mining on the Yelp data set, our team plans to begin our mining on smaller portions of the data set. Since we are creating a predictive model, we can compare our predicted user behavior with actual user behavior that occurs later in our data set. Using this technique, our group will be preforming cross-validation on our predictions to evaluate how accurate our predictive model can get.[6]

Our group can also perform validation on our test data set as it grows in size. We could see if our user behavior predictions grow in accuracy as the size of our data set being mined increases. This would tell us that our predictive model is accurate and would become more accurate with an increased amount of user data.

6. TOOLS

For our data mining project, our group plans to utilize several tools. We will use a NoSQL database to store our data such as MongoDB to store our data for easy access. Using this database will give our group a consistent way to access our data.

Python will be our main programming language for in depth data mining. This is an easy to use programming language that is great for data computation and analysis. A Jupyter Notebook can be utilized to run our python code on top of. This will allow us to run our code in individual cells which will speed up having to re-run small parts of code. Jupyter will also be used to display our final results and code in one, cleanly flowing display.

Python also has plenty of useful packages that can be used to simplify analysis. Numpy, ScyPi, and Pandas provide functions to help with numerical computation and provide our group useful statistical results. TextBlob will be used for calculating sentiment analysis of reviews. This library is simple to use and provides polarity values for strings of text. We will be passing in the review text strings through

this tool to get sentiment ratings.

To analyze the weather data we are exploring the use of the package "csvkit" csvkit "is a suite of command-line tools for converting to and working with CSV, the king of tabular file formats."[7] It has been helpful in not only pre-processing the data but converting it as well. It can convert the large weather database file to JSON with no error so that the weather can be manipulated through MongoDB as well.

7. MILESTONES

There are 5 major milestones for our project, each with a few sub-goals:

1. Integration of datas ets and tools - Once the Yelp data set and weather data set have been transferred into MongoDB they will be ready for manipulation. Also, the reconfiguring of the Heavy-Moose program will make text manipulable

2. Pre-Processing of data- Here we apply a number of preprocessing techniques on the data with the aim of reducing it to only that which predicts consumer and producer behavior:

a) Pre-Process and modify Yelp data set to include sentiment rating and also remove columns not relevant to user behavior.
b) Pre-Process weather data to be of manageable size and make csv available for use. The weather data came in a very large csv format with around 27 million data points. It is ten years of hourly weather data for multiple cities around the US. We will be narrowing our search to around a year(2016), and fewer cities. It is also reasonable for us to reduce the daily data to only times during and near open business. Finally, there are multiple columns that represent values that might not be as helpful to our search for predicting user behavior and we will be removing them.

3. Analysis of user behavior- After the data has been processed we can now perform a number of functions on it to determine the user behaviors that we hypothesized to see their frequencies and patterns. While we do have a method and ideas for how to search for user behavior, we assume that we will discover new patterns along the way. We intend on investigating promising new strategies we discover throughout the analysis stage. That being said, our overarching approach will stay the same:

a) Basic models of correlations between the data. Early on we will be building two basic models: The user sentiment rating-review score model and the weather coverage-review/sentiment model. We believe these models will show the intrinsic nature of users. This will provide a strong baseline for looking at how consistent users' are with their ratings.
b) Once we have discerned the baseline, we will be able to apply it in multiple ways to generate multiple new more complicated models. The aim of the more complicated models will be to wholly analyze some aspect of the user that is "interesting" based on our basic models.

4. Compilation of data- After we have identified frequencies and patterns they must be reduced even more into generalities that are presentable to Yelp, along with strategies we develop for using this compiled data:

a) Identify and attempt to explain any outlier models.
b) Combine refined models.
c) Produce literature relating our findings.

5. Presentation of data- our work must be condensed into presentation form for class, and into a form acceptable for Yelp. In our presentation we will show to what extent and how we can predict user behavior, and will describe the techniques we used to get those final answers.

7.1 Achievements

Our group has completed several of our milestones up to this point. In the integration milestone, we have taken all of our yelp data and transferred it into a MongoDB. The weather data set that we plan to use has been downloaded. After cleaning this data, it will be moved into a separate database from the yelp data.

The yelp data has been pre-processed to identify most active users. These users will be used as a test group for our user analysis. They provide a lot of useful data per user while not having to use data on all users to build a decent model.

Finally, our group has performed initial analysis on the data and gained some useful results. This initial analysis includes some basic and high level statistical analysis and the incorporation of sentiment data. We have been able to use this data analysis to make some conclusions about the yelp data. This is only preliminary and will require further data mining to find more conclusive results.

7.2 What Remains

While we are making great progress, we are still working on milestone 3.a. We are currently finishing up our analysis of the sentiment rating-review score model. We have found that sentiment by itself is not a good predictor for rating but are realizing that along with sentiment, average user rating and average restaurant rating might pair well to make a basic model. We need to create a model including these other factors, and we believe we will have a solid baseline.

We have acquired the necessary data to start the model that takes weather into account. The weather data has been mostly pre-processed, and we are ready to analyze it in the city of Charlotte, NC. The pre-processing we have left for the Charlotte data set is as follows:

a) Go back and look at the remaining columns
b) Come up for strategies on how to analyze these columns
c) Remove columns where an applicable strategy can not be found
d) Remove rows for hours where no data was taken
e) Remove rows whose time is not during average business hours (10PM-5AM).

We have chosen Charlotte because we believe its variable weather will be the most encompassing way to look at any possibility that weather is affecting user behavior. If we find that weather is affecting user behavior in some way, we can go back and analyze other cities to compare, but we are

assuming that the overall population will act in a uniform way. Other cities like Las Vegas would not be good cities to analyze because they have minimal variance in weather.

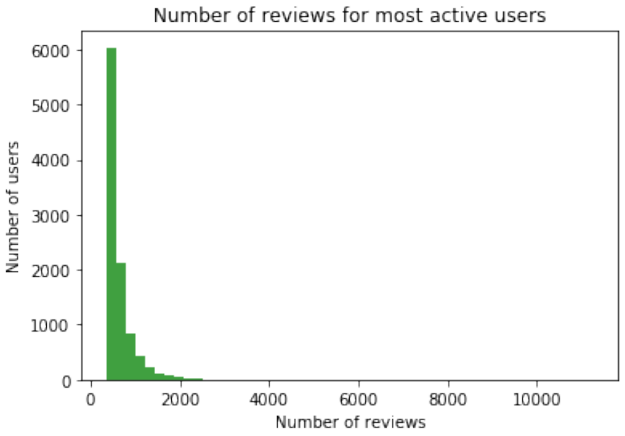
At the same time that we are building the weather model, we can still begin to build more complicated models concerning sentiment and rating score. We have made a reasonably deep analysis of this relationship and believe we understand it enough to begin adding other factors to see if we can predict user behavior to some certainty. These more complicated models can be added to the weather model in a variety of different ways.

Some of the MongoDB queries, and pandas data frame filtering and mapping are computationally expensive. It may be worth considering leveraging Big Data technology such as Apache Spark considering that Spark is very good at utilizing concurrency which will result in certain "map" functions utilizing multicore capabilities. Furthermore, this may be worth a consideration as a result of Python's Global Interpreter Lock (GIL), which makes true multithreading (with simple Python code) almost impossible. This is still simply an option, and if the overhead of setting up Spark is too much, than waiting through the Pandas computations will simply be a temporary burden. Regarding the MongoDB queries, it will be neccessary to go into MongoDB and set up "keys" such that the database optimizes itself in such a way around these keys such that queries on these keys become much faster. This will take analysis as to what keys will be neccessary for future queries. The combination of leveraging these two technologies could not only speed up computations, but allow us to perform even more refined queries without the need for waiting through the expensive computations.

Finally, we need to make our findings presentable. It is worth noting that as we continue to delve into this project, we are feeling more and more confident that we are going to be able to find a representable relationship between the data that can predict user behavior with some certainty.

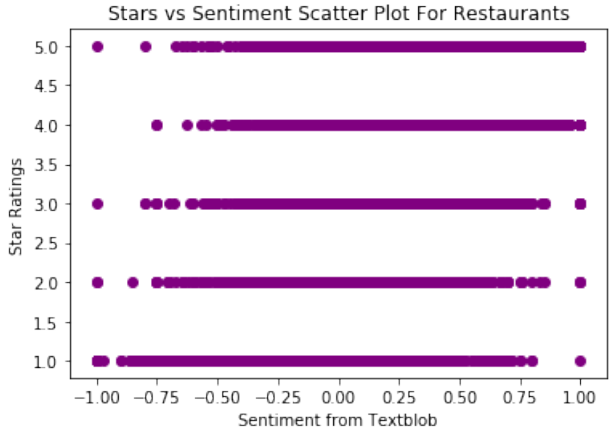
8. RESULTS

After setting up our tools and run time environment, we began collecting info from the yelp data. Three collections were created in our database named yelp which held the JSON data for business, user, and review. With our data more easily accessible through an interface, we could begin working with it. Using a python library to connect to our database, we began collecting data about our users. The first bit of information that we mined from the data was users with the highest review count. These users will provide the most useful data without using info from all one million users. In collecting data about the top 10,000 users, we found that most users have much less than 1,000 reviews while a select few have up to 10,000 reviews. This is shown in the graph below.



The next piece of data that we wanted to mine from this data set was a comparison between the ratings that users give in their reviews vs the language users use in their review text. A close correlation between these values will allow us to predict ratings from users based on the language they frequently use in their reviews. To do this, we first collected all of the star data that our selected users have given in some of their reviews. The star ratings were between 1-5 and were normalized to values between 0 and 1. Sentiment values about the text within each review were then calculated using TextBlob. The results from this comparison is shown in the plot below.

In this plot, there is a nice bell curve that the star ratings follow with most of the ratings being higher, around the 4 star mark. The sentiment ratings are a bit more condensed around the neutral to slightly positive mark. This tells us something interesting about our users. When given the option to assign a business a number between 1 and 5, users will generally be more aggressive with their ratings. The ratings, while most frequent at the 4 star mark, are more spread out and there can be several 1 or 5 star ratings from users. But, when users are given the ability to put their own words into a record, they generally don't use language that can be considered as aggressive. Using this info, we will be able to predict the rating a user will give to a business based on their sentiment and possibly a couple of other attributes.



The figure above shows a variety of critical properties of our data to take note of; specifically, it is a scatter plot of the star rating user's gave specifically for restaurants (the Yelp dataset consists of many businesses not just restaurants), vs the sentiment value calculated by the TextBlob module. First off, one can see that for a given star rating, there exist a variety of sentiment values. For example 1 star ratings range from -1 (very bad) to 1 (very good) sentiments. At first glance this is very nonsensical, as well as 5 star ratings. However, upon further thought, this starts to make sense. From a user perspective, a given user might not reflect completely through their language what star rating they will give. In addition, sentiment analysis is still a relatively complex field where individual words can significantly decrease or increase the sentiment calculations, while the general tone of the statement is either negative or positive. For example, one user's rating was "This place was really good when it was Social House, now its just an expensive place to get sub-par food... I would definitely try somewhere else and not waste my money here again." This is clearly a negative statement for most human interpretation. However, sentiment analyzers would see the phrase "really good" in the beginning and this would significantly raise the value. The sentiment rating was 0.169, which is slightly positive, but the rating was a 1 star. Moreover, as mentioned before, sometimes users aren't explicit in their reviews, swearing or all-caps (all good indicators for most sentiment analysis tools), yet still give ratings that can contradict this. An example of this is the following review "A convenient store with a good hot kitchen...(more review in between) ...You can either order a pizza or sub or you can take advantage of their hot and ready pizza by the slice. So stop by and grab a few slices of pizza and don't forget to get a Cold Classic Glass Bottled Coke to go with it." This review while in the middle mentions "recommend this place for the pizza by the slice" is a little above neutral (0.130) when interpreted by the sentiment analyzer, and objectively by many humans as well. However, the rating was a 4 star rating. Further analysis as to how to assess this information is provided in the next paragraph.

Even given this information, when calculating the Pearson correlation between ratings and sentiment, the value was 0.58, which suggests that there is a somewhat positive correlation between restaurant ratings and sentiment (the higher the sentiment the higher the star ratings). However, when performing an F-test for significance for sentiment in predicting what a given user is going to rate a restaurant, the value of the F-statistic was 99315.75. This F-statistic is exceptionally large, and the p-value corresponding to it was practically 0. It's important to understand that the null hypothesis in this case is that sentiment by itself is a good predictor of star rating. Therefore, a p-value of 0 rejects the null hypothesis that sentiment by itself is a good predictor of star rating. While the p-value and Pearson coefficient seem to contradict each other, its important to emphasize that the F-statistic is taken with respect to predicting star rating solely with sentiment. What this means then is that while sentiment may be a good predictor for star rating, it cannot be a good predictor by itself, if a classifier model was set up. More variables would need to be added to the model. Some other critical variables include the average rating for the restaurant, as well as compliments given to the reviewer including "useful" and "funny" compliments. The average rating for the restaurant would certainly be a useful predictor for what a given user is about to rate it, as assuming a large sample size (which is safe considering there are over 200000 restaurants in the dataset), what any given user is about to rate will likeley reflect the restaurants average, unless there are major changes to the restaurant in a given time period. Reviewer compliments are compliments given from other users to a given user that signifiy that the user has some recognition in the Yelp community. More reviewer compliments indicate that that a given reviewer is in a sense a "driver" or "leader" in the community and in a sense can influence how others rate a given restaurant. The next step would be to apply a classifier that takes into account these variables and then predicts a 1-5 star rating. One choice could be a K nearest neighbor classifier.

Table 1: Fundamental statistics about sentiment and star ratings

F-statistic	99315.75
P-value associated with F-statistic (4096 decimals)	0.
Pearson Correlation Factor	0.576

9. REFERENCES

[1]Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving Restaurants by Extracting Subtopics from Yelp Reviews." (2013): 1-5. Web.

[2]Kotzias, Dimitrios, Misha Denil, Nando De Freitas, and Padhraic Smyth. From Group to Individual Labels Using Deep Features. N.p., n.d. Web. <http://mdenil.com/media/papers/2 deep-multi-instance-learning.pdf>.

[3]Maurya, Abhinav, Kenton Murray, Yandong Liu, Chris Dyer, William W. Cohen, and Daniel B. Neill. Semantic Scan: Detecting Subtle, Spatially Localized Events in Text Streams. N.p., n.d. Web. <https://arxiv.org/pdf/1602.04393.pdf>.

[4]https://www.yelp.com/dataset_challenge

[5]Elias Bezanis, Oliver Hanna, Yang Wang
https://github.com/omnific-h/HeavyMoose

[6] https://www.ncdc.noaa.gov/cdo-web/datatools/lcd

[7] https://csvkit.readthedocs.io/en/1.0.1/