Yelp Mining Final Report

Tyler Lugger Amogh Jahagirdar Elias Bezanis

ABSTRACT

In this project, our group chose to mine data from a data set provided for free by the crowd-sourced review company Yelp. This data set contained a small snapshot of the different types of data that Yelp hosts on their site. The data contained within this set had millions of reviews from millions of users on thousands of businesses from a handful of cities.

Our group focused on answering two main questions. The first being, could we create a relatively accurate model which would predict user ratings based on their previous review activity? The second being, could outside forces beyond the restaurants control influence the star ratings people give to restaurants? For this question, we chose to use weather and do a direct comparison. Our results told us that we could predict with a reasonable amount of accuracy, what star rating a user would score a given restaurant. This prediction is very accurate for determining one star ratings and got slightly less accurate for higher star ratings.

The weather analysis showed us how average yelp star ratings in an area on a given day would compare to a weather score defined by our group. When these two scores are graphed together on a normalized scale we were able to find some correlations that would support our hypothesis.

1. INTRODUCTION

In this project, The main motivation of our group was to use the data collected by the review service Yelp to help restaurants that receive reviews. Yelp has been around for over 13 years 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 either help or hurt a restaurant receiving a review. 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 get their own experience at a restaurant. But bad reviews could drive potential customers away from the business without ever trying it for themselves.

We want to use Yelp data to give more power back to the restaurants being reviewed. This data is mainly used by potential customers of businesses to help them decide on the best restaurant to try out or the best contractor to call to fix their home. We believe that the data could be just as useful to the businesses in helping them optimize how they operate and bring in the most customers. By mining data about Yelp users that frequently use the review service, we planned to create a predictive model of user behavior. This would allow us to find out several, previously unused aspects about Yelp users and pass that information on to the restaurants themselves.

2. RELATED WORK

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. DATA SET

Our data set comes from Yelp directly. Yelp has selected a very small portion of their review data and made it available to anyone to work with for academic/educational purposes. This data set is nearly five gigabytes in size and contains millions of tuples spread across five JSON files. These files contain all of the information we will need about Yelp users, reviews of businesses, and information on the businesses themselves.

Since Yelp has made this data set part of a challenge, our team will have the opportunity to submit our data mining results to the challenge for an opportunity 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.

4. 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.

5. TECHNIQUES APPLIED

Using multiple data sets for our mining project required us to apply several data mining techniques. The following subsections outline many of the techniques used by our group.

5.1 General Pre-processing

In using two different sets of data, both the yelp data set and a weather data set, our group was exposed to a variety of data mining techniques needed for the data. The Yelp data set provided great data from the review hosting company that required minimal cleaning and processing. There were several JSON files provided with info about users, businesses, reviews, check-ins, and even pictures from the different reviews. Initial cleaning only required us to choose the files that we wanted to work with. Three of these files seemed the most applicable to the mining we planned to do. We chose the user, review, and business data files and placed each of these into a Mongo database where each file made up its own collection in the same database.

From this point, we could easily access data from the database on our users, businesses, and reviews. But in grabbing different attributes and applying processing to our data, it became very apparent that we would be unable to work with all four million lines of the data set when developing our model. To fix this, we chose a smaller subset of data from the database and chose it as our test set. This subset ended up being the top 10,000 users with the most reviews. This would allow us to work with a smaller number of users but still have a large amount of content to work with.

5.2 Classification of User Reviews (User Behavior)

One of the topics that we were interested in was user behavior. Specifically, one question was whether it was possible to predict what a user was about to rate. Since star ratings are discrete values, and our goal was to predict these discrete values, a classification algorithm was required. Two possible classification algorithms were the K-nearest neighbor algorithm as well as the Support Vector Machine. It was shown with an F-test that sentiment for a given review text could be used as part of a classification system to predict the star review, however, the sentiment just by itself would not be enough.

At this point it was known that not all features would be useful, and could even potentially skew a model.The K-nearest neighbors is sensitive to outliers, as well as bad features, while the SVM handles both of these quite well. There were two routes to consider. Either perform principal component analysis, so that the most prominent features were extracted, and then apply the K-nearest neighbors, or simply just train the support vector machine. Since the data set was quite large, performing two heavy computations for K nearest neighbors was deemed difficult, and the support vector machine was trained instead. Perhaps, given a toolset consisting of Big Data tools such as Spark, it would be possible to efficiently train a K-nearest neighbor, however for the sake of prototyping a SVM was used.

Sampling techniques were also used in the process of training a SVM model. There conisted 200000 rows of data. Training using all this data would not only be computationally expensive, but be heavily biased as there was no testing data. As a result, there needed to be a medium to split the data into training and testing data, while being computationally expensive. Using sampling statistics this issue could be solved. The procedure was as follows; sample 25000 rows in the data, for training the SVM. Then sample 10000 rows for testing the SVM in the current instance. Repeat this procedure 10 times, and while repeating the procedure calculate the accuracy for each star rating during each iteration. At the end the accuracies for classifying each star rating would then be averaged. This sampling technique allowed us to efficiently train the model, while still being statistically accurate.

5.3 Weather Analysis Techniques

Using a weather data set required our group to apply better cleaning and pre-processing techniques. In getting the data set, our group first needed to choose where this data set focused on and when this data occurred. This decision was easily made by looking at the cities that we had available from the Yelp data set. From the 11 cities in the data set, our group chose to focus on weather and reviews from Pittsburgh, Pennsylvania. This location would provide us with plenty of variable weather patterns. Using analysis of this along with the Yelp reviews in this location, we could easily see if review scores were at all influenced by the weather. Some of the cleaning from this data set that we still had to do was stripping weather data that would not be relevant from us. We only needed weather data during business hours of restaurants. This would significantly reduce our data set size and keep only relevant values that would be used. This data set was kept in a plain csv file for processing.

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.

6.1 Jupyter Notebook and its Advantages

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. This is particularly useful, since many of our operations include expensive queries on MongoDB. These MongoDB queries can be multilayered on millions of records. Storing the results in a cell will remove the need for duplicates of these expensive computations. Jupyter will also be used to display our final results and code in one, cleanly flowing display.

6.2 Python

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.

6.3 R

To train and test a support vector machine classifier (SVM), the programming language R will be used. R is a statistical programming language, and also has the concept of cells in a notebook. R provides a library called "e1071" which has a SVM trainer, where it is simple to pass training data frames to the classifier and generate a model. R and Python are both scripting languages, that allow for rapid prototyping and for more focus being applied on the data analysis as opposed to the entire software engineering process.

6.4 MongoDB

Another tool that is being used is MongoDB. MongoDB is a NoSQL database, which is specifically document oriented. Binary JSON is the underlying data structure for storing records. This was used specifically since the Yelp Data given

was in the form of JSON files. As a result, all that was needed was to use them mongimport command on the JSON files, and they were automatically loaded into collections in MongoDB. This would save us the time to write parsing code that would parse the JSON and insert it into a MySQL Database. MongoDB queries were initially slow, however, after establishing keys on certain attributes of the data it became significantly more performant.

6.5 CSVKit

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:

7.1 Integration of data sets 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.

7.2 Pre-Processing of data

Here we apply a number of pre-processing 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. For classification of restaurant reviews, it was neccessary to first filter out restaurants from all the businesses given. Then, the resulting restaurant ids were used in order to query all reviews from all of these restaurants. Then F-tests, and statistical metrics were used in order to filter out the unneccesary columns.

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.

7.3 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.

7.4 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.

7.5 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. Our presentation will also showcase our results from weather analysis. We will show our findings for weather influencing user behavior in different ways an how we interpret the data.

8. RESULTS

Our group has completed all of our milestones at 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 used was downloaded from NOAA and properly cleaned and pre-processed. This processed data was saved as a new csv file and was used during the analysis of our user reviews vs weather.

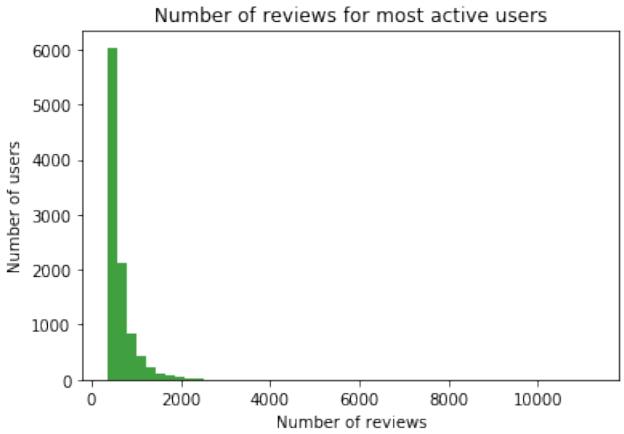
The yelp data has been pre-processed to identify most active users. These users were a test group for our user analysis. They provided a lot of useful data per user while not having to use data on all users to build a decent model.

Our group began by performing initial analysis on the data. From this we gained some useful results. This initial analysis included some basic and high level statistical analysis and the incorporation of sentiment data. We were able to use this data analysis to make some conclusions about the yelp data. This is was preliminary and required further data mining to find more conclusive results.

8.1 Initial User Activity Analysis

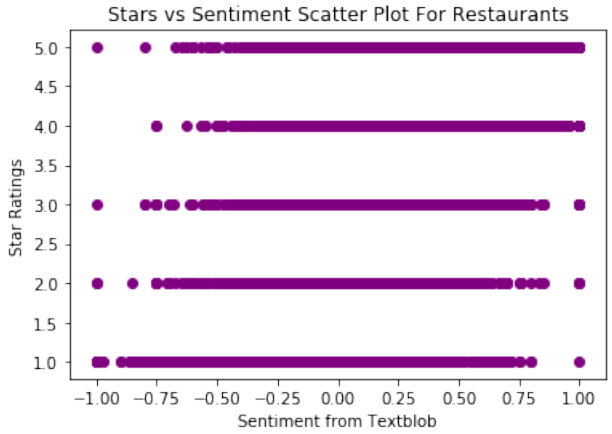
After setting up our tools and run time environment, we began collecting info from the yelp data. Three collections were created in our database named yelp which held the

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

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

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 data set), what any given user is about to rate will likely 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 signify 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 or a support vector machine.

8.2 Predictive User Model

As mentioned before, it was discovered that sentiment may be a good predictor for star rating, however it was certainly not a good predictor by itself. There must be some other variable that can be used in order to predict star rating. Since, a given user's rating is likely to be close to the historical average for a restaurant, this metric could be used. In other words, a single user's review is unlikely to deviate far from the average of the given restaurant's star rating, which is calculated by averaging stars across all reviews for a given restaurant. This data was provided to us in the Yelp business data.

Now, two factors could be used in order to predict what a user is about to review a restaurant; the sentiment of the review text, calculated from TextBlob, as well as the restaurant's average star rating could be used. Since, the goal was to classify this user behavior of rating a restaurant, trivially, a classification algorithm was necessary. Two algorithms were considered, the K-nearest neighbor (KNN) algorithm and the support vector machine (SVM). After research, the SVM was chosen for a variety of reasons. First off, KNN is exceptionally sensitive to outliers. Since, as mentioned before, there exist many outliers particularly with regard to sentiment. Moreover, the KNN algorithm is sensitive to variables that do not account for statistical variance in the predicted value. In other words, if there are irrelevant columns in the training data the KNN algorithm will not perform as well.

While, irrelevant data was filtered out with F-tests, it may have been worthwhile to go back to the original data set and perform PCA (principal component analysis). Since PCA on a dataset of size 200000 would be computationally expensive, the best route for prototyping a model was to use the support vector machine.

8.3 Sampling Statistics for Building Predictive User Model

Using the sampling method described in section 5.2 of this document, a support vector machine model was generated as well as an average accuracy for predicting a given star was generated. Table 2 details the specific numbers regarding how well the SVM model correctly classified a star. As one can see, the SVM classifier accuracy for 1 star is extremely high at 92.83%. There are a variety of reasons for why this could be the case. One hypothesis is that very poor sentiment is easily picked up by the TextBlob library, and as a result the SVM is more easily trained to correctly classify when a user is about to give a 1 star rating from the text. Another hypothesis could be that 1 star and 2 star (which had another high accuracy of 83%) restaurants generally have a high concentration of users that dislike the restaurant. As a result, the average restaurant star rating is a very good representative of the population. Therefore, the model would have better accuracy predicting that. Another interesting thing to note is that 4 and 5 star ratings, have the lowest accuracy of 57% and 60% respectively. This simply could be the opposite case for why 1 star accuracies are higher; that is to say, it may be more difficult for TextBlob to accurately assess high sentiment for a given review text. In addition, a restaurant may have on average a 3-4 star rating, yet the sentiment may have high value, and as a result them model may incorrectly classify it as a 5 star

Table 2: SVM Classifier Accuracy for a Given Star

1 Star	92.85%
2 Stars	83.68%
3 Stars	76.18%
4 Stars	57.03%
5 Stars	60.3%

rating. These are simply hypotheses, and in order to truly delve into this question with statistical backing, it would be neccessary to involve the use of more data as well as more advanced tools.

8.4 Weather Influence

One of the questions that our group sought to answer was, are user ratings of restaurants influenced by factors outside of the restaurants control? To check for this, we chose to compare how different weather conditions could affect the star ratings that users would give to a restaurant.

To do this analysis, our group first began by plotting weather conditions against star ratings. Additional processing was needed for both the yelp and weather data sets to create this plot.

First, our group had to collect a weather data set with relevant weather information about one of the cities available to us in the yelp data set. We chose to use a data set for weather in Pittsburgh in all of 2016. We chose this city and time because it would provide us with variable data during a time where we had a good amount of yelp reviews to use for analysis. The weather data was cleaned in the ways mentioned in section 5.1. Empty values were removed and the data set was filtered to only include weather data during the times that restaurants would typically be open.

This data was then run through a python script that would collect all of the weather information for a given day and assign a predetermined weather score. This score ranges from -1 to 5 which is a range of bad to good weather conditions. The bad and good weather conditions are based on what our group believed the average person would consider "bad" or "good" weather. The plan was to derive this "average weather rating" and compare it in a variety of ways to the average rating for a certain day.

The way the weather rating was created was based on two columns from the weather database. The first column that we used was the 'HOURLYSKYCONDITIONS' column. This column contains the cloud coverage during any given hour. The type of cloud coverage was represented in by a variety of different codes. The database came with a document prescribing weather conditions to each code. With the 'HOURLYSKYCONDITIONS' column we created a base value for the weather rating. Then, the other Column we used was 'HOURLYPRSENTWEATHERTYPE.' This column had the type of weather that was occurring at any given hour. Based on the type of weather that was occurring with any given cloud coverage we would modify that hour's weather rating appropriately for the given weather type. The precise mapping from weather to rating is as follows:

Table 3: Hourly Sky Conditions(Coverage)Base Weather Rating Base Values

Clear = 5
Few Clouds = 4.5
Scattered Clouds= 4
Broken Clouds= 3.5
Overcast=3
Light Rain=2.5
Foggy2.5
Heavy Rain=2
Thunderstorm= 2
Varible Visibility(very low vis)= 2

Table 4: Hourly Present Weather Type Base Weather Rating Modification

Fog = rating-1
Rain = rating -2
Snow= rating-2.5
Hail= rating-3.5
Mist= rating-1.5
Drizzle= rating-1.5
Snow Grains= rating-3

So for example if the weather was overcast(Table 3) our could would initially produce a rating of 3 for that hour. Then the we check the other row, 'HOURLYPRSENTWEATHERTYPE,' to see if there is any variable weather happening during that hour. If it were raining that hour(Table 4) we would subtract to from the value three to get an hourly weather rating of 1 or "poor." The group figured this numeric scheme was a great way to represent the change of weather visually, and also a good avenue to attempt to connect a rating to the current weather. We worked hard to come up with what we believed to be a "balanced" conversion scheme from weather to number.

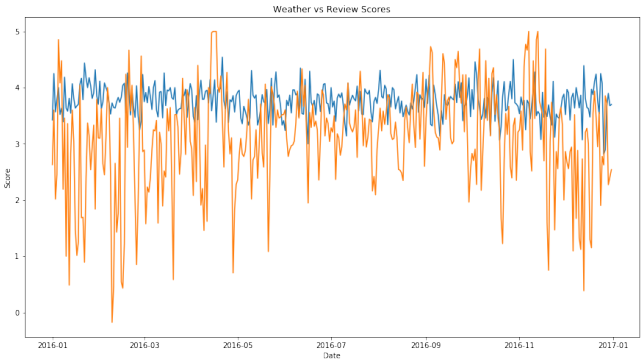
There are a number of ways that the analysis could have been more effective. For a more inclusive weather rating we could add more columns. For example, there is a wind speed column. If we decided, or it became apparent, that large wind speeds is generally considered "bad weather" we could have further tuned our value for weather rating based on wind speed.

For stronger and more wholesome analyses the yelp reviews would have to be hourly and we could have used a bigger data set. If the Yelp reviews were hourly we could be mapping the direct weather that a user was experiencing to their review as opposed to the average which feels and behaves more like a guess. Also, it is worth noting that all our weather data came from the same place in Pittsburgh and is not an average from all restaurants. Weather is not the same through out a city, and therefore it is reasonable that the "technical" weather rating was different at the place where the rating was given vs. where the weather data was taken. That being said, just getting a weather station inside the city is about the most accurate/wholesome information obtainable.

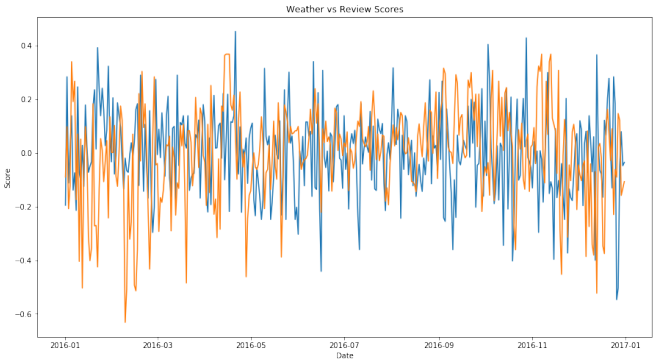
The next step was to process the yelp review data to give us

a single star rating for each day in 2016. First we needed to focus our reviews on restaurants from Pittsburgh. This was done by first querying the business data collection and grab business ids for all businesses in Pittsburgh. This result was then stored in a series and used in the query to the review data collection. This query grabbed all reviews in our series of businesses in Pittsburgh. We only grabbed reviews that occurred in 2016 and were tagged under the "Restaurants" category. this provided us with 10,000 reviews to use for analysis. The reviews were aggregated by date. For each day of 2016, the mean star rating for all reviews in Pittsburgh was found and mapped to the individual dates.

We now had all of the data needed to create a comparative plot. We started by simply plotting our weather scores vs date on the same plot as our average reviews vs date data. The following plot was produced.



This plot gave our group some confidence behind our predictions. With weather scores in orange and average ratings in blue, we were able to find specific points where weather scores and average star ratings appeared to line up. Placing both of these data plot on the same axis would provide us with a better plot to use for comparisons. To achieve this we took the averaged star ratings and the averaged weather scores and normalized them. After plotting them again, we were left with this plot.



This plot provided us with even more confidence for our hypothesis. While is is not perfect, there is a noticeable correlation between weather scores and average star ratings for all of 2016 in Pittsburgh. This tells us that while the data is not conclusive, it is absolutely something worth looking into further and doing further data mining on.

This analysis went slightly further to check for the changes of review scores based off of the changes of weather scores. We first began by counting the number of instances where the weather score had decreased from the day before. This would tell us that weather was unusually worse because it had been nicer the previous day. We then took the number of instances where average review scores and weather scores were lower than the previous day to show when people had worse opinions about restaurants during worsening weather. Using this data our group found that 49% of the time, if weather conditions decreased, so did average review scores.

The same analysis was performed for weather conditions that were better than the day before. This would give us a count of the number of days in 2016 where weather conditions had improved. Combining this with the number of days that average review scores had improved gave us a 46% correlation.

Roughly half the time, the average review scores follow the same change as weather scores. While this does not definitively tell us that user review scores are influenced by weather in some way, we can still say that weather influence on user review ratings is a topic to be studied more in depth. A more thorough analysis would allow our prediction to be proven or not.

Our group strongly believes that if Yelp had hourly review timestamps instead of daily we could have more thoroughly analyzed this question. We believe we took solid steps in the direction of answering the question of weather affecting rating, but the data isn't there right now to draw a supportable conlusion. That being said, we feel like there is definitely some connection between the weather and an average review score, but we can't claim to what extent this connection operates. If our results were more substanative, it would have been possible to add weather into our model when predicting user ratins.

9. APPLICATIONS

It is important to consider the business implications derived from the analysis generated during this whole process. Data Mining serves a vital purpose in making business decisions, such as optimizing procedures, or improving customer experiences. In the following paragraphs, several ideas will be discussed regarding how Yelp, and the businesses that are reviewed under Yelp can improve customer experiences as well as increase revenue benefiting both their shareholders and customers.

First, the application of being able to predict what a given user is about to review a company, using just the company's historic average rating as well as the sentiment value of the review text, will be discussed. To improve customer experience for the Yelp mobile application, a classifier could use the two aforementioned variables to accurately predict the star rating the user is about to give. After this prediction, it could "auto-fill" the stars. For example, if the classification algorithm predicts from the sentiment of the text and the given business's historic rating that the user will rate a 2 star rating, then on the mobile app instead of having the user input the stars, 2 stars would automatically be filled. This may seem trivial, but many Yelp reviewers do want

others to view their opinions, and as a result the reviewers can potentially put a lot of thought in to the star ratings. Auto-filling the stars, while a very simple change, can alleviate some of the burdens from the reviewing process.

Since the aforementioned process expedites the time needed to input a review, this could also increase the volume of reviews sent to Yelp. This would allow 2 things. First, since more reviews are being sent to Yelp, they can use this larger amount of data combined with their Big Data infrastructure, which includes tools such as Apache Spark and Cassandra, to train more complicated models as well as improve the original models. More complicated models could include concepts such as deep learning using neural networks. Next, a larger amount reviews, would theoretically mean that more "favorable" businesses would shine, and very "unfavorable"(1 star) businesses would decline in value. This would allow shareholders for these companies evaluate their portfolios, and distinguish successful as well as poor companies.

10. 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/2015-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/