Data Analysis using Yelp Dataset

Engineering Big Data

* Vineet Dandekar
* Saurabh Rao

**Introduction –**

Yelp was founded in 2004 to help people find great local businesses like restaurants, dentist, hair-stylist, etc. In Q4 2015 alone, Yelp had a monthly average of 86 million unique visitors. The yelp users have written more than 95 million reviews by the end of Q4 2015. In addition to reviews, users can find events, list and talk to other users. Every business owner can setup a free account to post photos and message their customers. All these features on Yelp produces a lot of data.

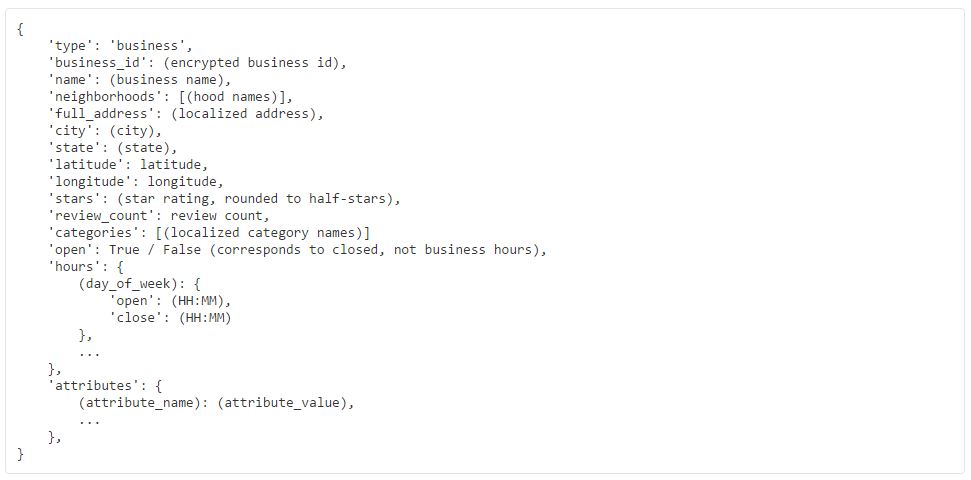
Every year Yelp makes this data available on the Internet as a dataset challenge. Yelp wants us to make use of this data in an innovative way to find interesting trends and patterns. Being regular users of Yelp.com, we found this dataset very interesting and decided to take up the challenge set by them and hence, used this dataset for the Engineering Big Data Final Project.

**Dataset –**

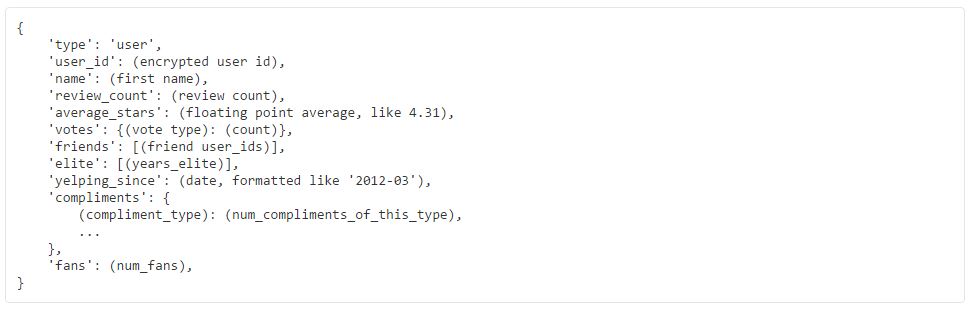
The entire dataset provided by Yelp is approximately 8GB, consisting of six files in JSON format and a huge collection of photos. The files have data of 2.2M reviews and 591K tips provided by 552K Yelp users for 77K businesses. Each business has its related attributes like hours, parking, availability, ambience and many more. The dataset also provides social network of 552K users for a total of 3.5M social edges and aggregated check-ins over time for each of the 77K businesses. 200K pictures related to different businesses are also provided.

In our project, we have made use of mainly three files, viz. business, user and review. All three files are in JSON format and have the following structure.

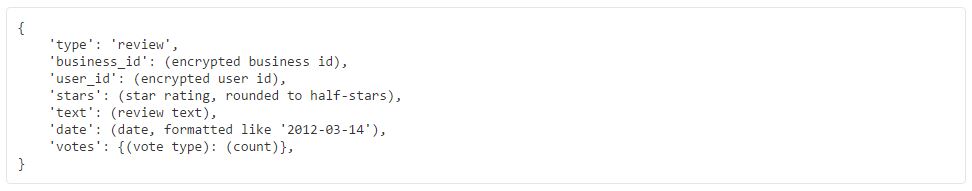
**Business:**



**Users:**

****

**Reviews:**

****

**Problem Statement –**

The business data from Yelp consists for all the businesses which not only contains information about restaurants but also other businesses like dentists, grocery stores, garage, etc. In our project, we have only performed analysis by considering businesses of the category restaurants. All are analysis and prediction models based on this type of business only.

When we consider such businesses (businesses falling under restaurant category, and here after a business will always mean a business of the type restaurant), the rating which is provided to them is most often than not based on the type and/or quality of food served. But, when we read the reviews for a business, we can see that many users comment on other features or services provided by the business. This made us believe that there can indeed be other factors as well, which are responsible and can affect the ratings for a business. Based on such belief and some background analysis on the business data, we decided to create a model which can predict the success rate for an upcoming business based on factors other than the food served in a restaurant, such as the facilities provided like parking, delivery, reservations, internet or the ambience and noise level, etc.

In addition to that, when we analyzed the user reviews provided to the businesses, we found that in many cases the ratings which the users provided was based on just a single factor which the user liked or disliked. We concluded that the ratings provided by the users for a typical business was not always a correct measure of the actual quality of the business. Hence, we decided to write an algorithm that can automatically calculate a rating based on the sentiments contained in the review provided by the user.

Lastly, we developed a model which can efficiently recommend a business to the user based on his/her previous activities and preferences.

**Technologies –**

* **Big Data Ecosystems:** Hadoop 2.7, Mahout
* **Database:** Hive
* **Tools:** Microsoft Azure Machine Learning Tool, Qlik Sense
* **APIs:** IBM Watson Alchemy, Hive-JSON SerDe, Hive-JSON Schema
* **Programming Language:** Java 8
* **IDEs:** NetBeans 8, Eclipse
* **Platform:** Windows 8, Ubuntu 14.04

**Data Loading & Analysis –**

As mentioned earlier, the dataset was entirely in JSON format and the structure of the JSON element was very complicated. Each JSON element had a complex hierarchy which was not uniform over all the elements in the file. This caused us a lot of problem for storing the data and retrieving only the information which we wanted. Most of our time was spent in figuring out how we could save the dataset files so that it would be easier to retrieve data for analysis. We tried writing custom Java code using JSON Parsing APIs, but the structure being very complicated, the code did not work efficiently enough for our liking. Then we tried writing scripts in Python and R but failed to achieve what we were expecting. We even tried converting the JSON files to CSV by using codes from the internet, but the huge size and the complicated structure of our data made even those codes to fail.

After series of failures, we decided to use some big data technology. We tried Apache Pig to directly access the JSON structure, but again the complexity of the JSON structure made it very difficult for us to write Pig scripts to retrieve data we needed. We tried several other technologies before finally settling with Hive. We found that we can directly dump the entire dataset in Hive.

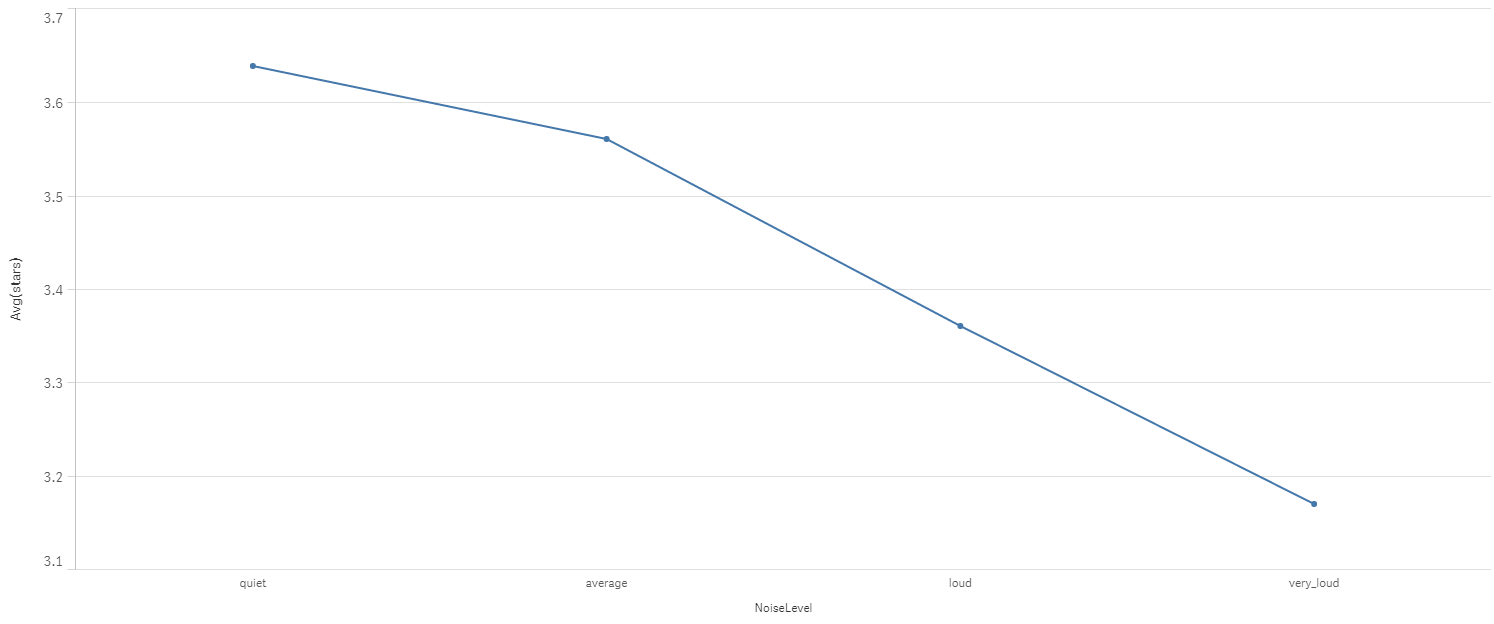
But, what seemed to be a very easy task of just configuring Hive with Hadoop HDFS and then storing the data inside Hive turned out to be a lot more tedious. We found that Hive does not store data in JSON format with its basic configuration. Hive needs a serializer/deserializer program in order to do what we were trying to achieve. So, we used a JSON **SerDe** API (<https://github.com/rcongiu/Hive-JSON-Serde>) which performed the serialization and deserialization tasks for Hive, which in turn made it possible for Hive to store the JSON data.

Hive is just like a relational database where it stores data in the form of tables with columns and rows. So, in order to store the data, we needed to provide a schema in the form of a Create Table query for creating each table. Writing a schema for each table was turning to be a tedious job due to the mere complexity of the JSON structure. As a result, we made use of another API which creates a Hive schema based on the JSON file provided to it (<https://github.com/quux00/hive-json-schema>). This API made our job very easy and the entire dataset was loaded into Hive in no time. After that, retrieving data from the Hive schema was just a piece of cake. We just needed to fire Hive queries (which are almost similar to SQL queries) to retrieve the desired data and write it in files. This retrieved data was used for analysis purpose.

We performed analysis on the business and review data to look for some trends and patterns which were substantial enough to affect the ratings for any business in a positive or a negative way. Many of such analysis led us to develop our first model to predict the success rate of an upcoming business.

**Model 1: Model to Predict Success Rate of Upcoming Businesses –**

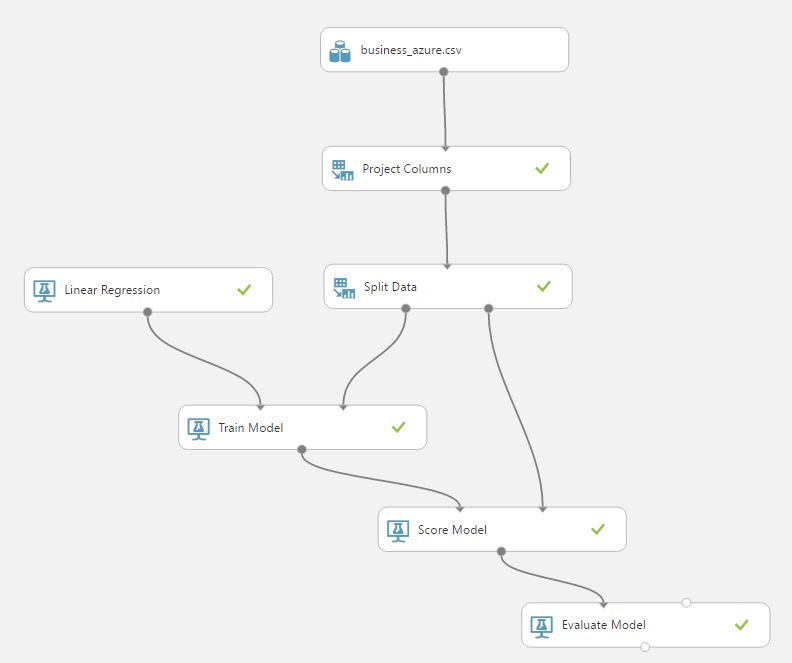
On the statistical analysis which we performed on the business data, we found many patterns and trends in the data which led us to believe that there indeed were many other factors other than the food served in the restaurants which were affecting the ratings for the businesses. Obviously, not all factors had a significant role, but some of them showed some definite and visible effects on the ratings. One of such factors was the noise level attribute which was associated with each business. There were four levels of noise in which the businesses were categorized into, viz. quiet, average, loud and very loud. When we plotted a line graph to show this trend.



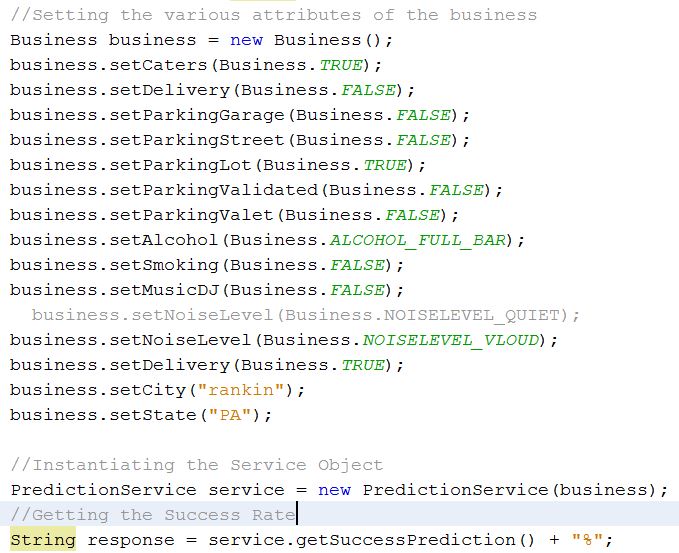
From the above graph, we can clearly observe that as the noise level goes up, the average rating for the business goes down. This graph was plotted using Qlik Sense. There were many other such factors which played an important role in affecting the ratings for the business.

With these observation, we came up with the idea of developing a model which can effectively predict the success rate of an upcoming business based on the factors and/or services which the business owner is willing to provide. This model would take the various inputs like the services the business would provide such as delivery, take-outs, reservations, drive-thru, internet, parking facility, etc. It would also consider the ambience, the noise levels, whether alcohol is served or not and many other such factors. Considering all these factors provided as input, it would give out the probability of the business idea to succeed in a desired location. One thing to keep in mind is that this model would only consider than services and other factors provided by the business and does not consider the quality or the type of food served. This model can only help the business owners to identify what all features should be included in their restaurant and which things can hamper their business.

The predictive model was created using Microsoft Azure Machine Learning Tool. The data provided to the model was the entire business (only businesses under restaurant category) dataset. Even if the entire table was provided, not all columns from the data table were used to predict the rating. Along with the various attributes for the business, the dataset also had information about the number of reviews written for a particular business. But, this column would not be necessary to predict the success rating for a new business, because initially a new business will have zero reviews and if number of reviews become a factor, then the success rate would not be calculated correctly. Many other such irrelevant columns were ignored. Only the columns which we considered as the features or services for a business were considered and sent to the training model. Below is the screenshot of the predictive model which we developed.



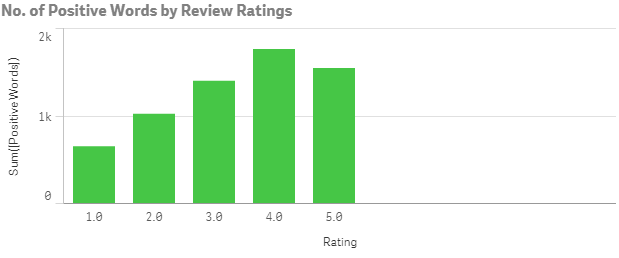
We provided 75% data to the training model and 25% to the testing model. After the model was trained to provide the success rate, it was hosted as a web service which could be consumed externally. In order to do that we created a Java project which could consume the web service and give out the success rate for the business. This Java program can be included as an external library class to use the model.

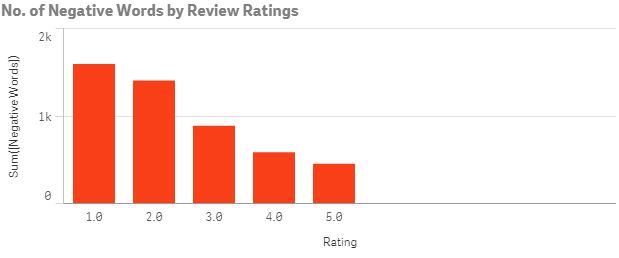


The practical use of this model can be done by Yelp themselves. They can include a feature in their application wherein a user can actually provide the attributes for the new business he wants to start and check whether it would be successful or not.

**Model 2: Model to Predict Rating for a Business based on the review –**

Along with the business data, we also analyzed the review given by the users for the businesses. We wanted to get the user sentiments contained in the review and for that we calculated the number of positive and negative words appearing in each of the reviews. There were some significant observations we found from the analysis.





From the above two graphs, we can see that as the rating increases the number of positive words increases and the number of negative words decreases. This is something which is very obvious that more the ratings, more will be the positive sentiments and less the ratings more will be the negative sentiments. But, however obvious it may seem, we cannot ignore the fact that there exist positive words even in low rated reviews and negative words in high rated reviews. This analysis made us to believe that there exist some positive feelings even in a review which seemed to criticize the business. As a result, we manually analyzed some of the user reviews and found out that generally the user tends to rate a particular business on the basis of one or some particular aspects which he liked or disliked and does not take the overall aspect of a business while rating it. For example, a user writes a review “I went to this particular restaurant. The food was good, but I did not like the ambience in the restaurants. Also, it was very difficult for me to find a parking space and hence I had to park my car a mile away and walk all the way. That was very frustrating...!!!” and gave a one-star rating. Now, if you can see, he claimed that the food was indeed good but he did not like the ambience and the parking facility. This suggests that he gave the rating based on just two factors ignoring the fact that the food was good.

This is one of the problems yelp is facing. The user ratings are actually not a correct measure of the reviews which the users provide which we can say is not fair for the businesses. Thus, we created an algorithm which will automatically predict a rating based on the sentiments in the reviews provided by the users.

For analyzing the sentiments, we used an API provided by IBM-Watson named as IBM-Watson Alchemy API. This API returned keywords with a sentiment score from a sentence or a group of sentences. The score ranged from -1 to 1 wherein the score tending to -1 had negative emotions, score tending to 1 had positive emotions and score tending 0 had neutral emotions. Based on all the sentimental scores received, we made some internal calculation to predict a star rating to the sentence (here, a review).

**Model 3: Model to Recommend a Restaurant to a User**

The third model which we created was a Recommender Model to suggest restaurants to users based on the user preferences and previous activities. We used the Mahout Library to implement an algorithm which would do the recommendation job. We used Java to code and build the algorithm wherein it would take the user ID as the input and as an output it would suggest four restaurants. The number of recommendations is customizable.

**Future Scope –**

The first model was created based on the data present in the business dataset. We can analyze the reviews and guess what type of facilities are required for a business to be successful in a specific location.

We can also make you for the user review to make recommendations based on the reviews submitted by the users by guessing what the user preferences are, and what the users like or dislike.