# Yelp Dataset Challenge

**Group Number 1:**

Vanessa Klotzman

Olga Kupchevskaya

Jeff Yoshida

Megan Meyers

December 5th, 2017

Data Mining, COMP 541

Professor Wang

## Table of Contents

## I. Objectives

Our primary objective was to determine whether or not the business attributes given in the dataset influenced the success or failure of a restaurant found on Yelp (i.e. business is currently operating at time of data capture). Secondary objectives included discovering the influence of reviews, including specific words in reviews, the optimal number of stars that indicate success (we discarded the automatic assumption that it would be 5), business success by location, the reasons why businesses fail, how images influence the business success, and creating a predictive model for determining if a business will succeed or fail. We also hoped to discover hidden patterns in the data using unsupervised learning techniques.

## II. Background information

Our group decided to participate in the 10th Annual Yelp Dataset Challenge (www.yelp.com/dataset/challenge). The challenge involves researching and analyzing the Yelp Business Dataset, which provided the following categories of attributes to examine:

- Business Stars
- Business Reviews
- Business Categories (e.g. italian, fast food, pizza)
- Business Attributes (e.g. accepts credit cards, wheelchair access, bike parking)
- Business Location
- Business Details (e.g. hours of operation, days of week open)
- Business Check-ins (users “check in” with the Yelp app when they visit the establishment)
- Photos taken of the business
- Tips left by Users (approximately 1 line reviews)

The dataset included both categorical (nominal) and discrete numerical data, as well as text and images. The original size of the database was approximately 10 GB, built by SQL script into Google Cloud Platform Cloud SQL (Google Cloud SQL Doc).

### A. Preprocessing:

We applied three of the four major data preprocessing steps  to this dataset in an iterative fashion. Data integration was not needed as this was historical data from one source (Han 155). A chronological reporting of our process is as follows:

1. *Data reduction*: We decided to work only with food-related businesses in the US. We manually discarded all businesses whose category had nothing to do with food (doctors, auto mechanics, etc.)  Businesses with NULL categories were also discarded. Since the foreign key relationships had to be broken in order to change the sql database in the manner which we saw fit, all orphans of these deleted businesses were manually deleted.

2. *Data cleaning*: We checked for and fixed inconsistencies in the data (Han 144) such as a state value having a number when it should be something like “CA” or a zip code with 4 digits instead of 5. Since the data is pulled in from a computer/web application, the data-entry conventions placed by the programmers prevented most kinds of problems with inconsistencies in most of the data. This allowed us to manually fix these problems since it was a very tiny set.

3. *Data transformation*: Upon learning that PCA would be a requirement for this project, we made additional columns for each of our numeric values, normalized the numeric values with a min-max normalization formula (Teetor 215) for the new min of 0 and the new max of 1, and inserted the values into the new columns. This way we were able to keep our original numbers and still  be able to use the new normalized values only when we saw fit.

4. *Data Cleaning*: We checked through all of our new normalized values for problems and found none.

5. *Data transformation*: We realized that the format of some of our attributes would not lend itself to statistical/mathematical models (De Jonge 23). For example, we had a column of attributes with nominal values (i.e. outdoor seating, allows dogs, smoking allowed) which we used one hot encoding on to expand the data into additional columns with a 0 for “attribute does not apply” and a 1 for “attribute applies.” Originally we were going to just a do a dummy variable with 35 levels (for 35 distinct attributes), but because many businesses have multiple attributes, and because this introduces data redundancy into the models (James 176), we decided one hot encoding was the best way to deal with it, despite the fact that we are introducing (temporarily) more dimensions (they will be reduced later via reduction algorithms). This applies to the category variables (i.e. restaurant, food-stand, mexican, pizza), as well, and we are using the exact same encoding for the exact same reasons with the category attribute.

6. *Data cleaning*: After one hot encoding the attributes we noted several null values had been introduced. We deemed it acceptable to find all NULL values and replace it with a ‘0’ (attribute does not apply) in this case. Later, the reason for the these NULL values became clear during another overall cleaning session, and the businesses were removed. (We had missed removing some non-food businesses - this is an ongoing problem with this dataset, which is why we clean so often).

7. *Data transformation:* The format of several other variables in our database were problematic. For example, in the hours table, the companies’ hours open information was all compacted into into one column:  Monday|10:00AM - 11:00PM. We used an R split algorithm to turn this into 3 columns, one for day, one for open time, and one for close time. The day was encoded as a dummy variable and the times are just 10 for 10 AM and 15 for 3 PM using the 24-hour military time format. At that point, we were still considering if this way the best way to do this and whether or not we had to come back to this at a later time and change it.

8. *Data transformation*: Because of the previous step database had 10 tables, and we were facing a lot of joins whenever we run a query. This was  HIGHLY problematic for reasons of both slow computation and tremendous data redundancy. Therefore, we moved denormalized, aggregated table into BigQuery (Google BigQuery Doc), thereby avoiding and/or reducing  joins.

9. *Data cleaning* - After aggregation and denormalization, NULL values have been introduced into the aggregated data. Nulls are filled in as follows: Missing binary attributes filled in with 0 (attribute and category).Missing total hours and total days information filled in with attribute mean after na.omit. Missing tip or checkin counts filled in with 0 (James 100).

10. *Data reduction* - categories reduced by dropping all categories with frequency less than 100. This reduced the dimensions from 250 to 135.

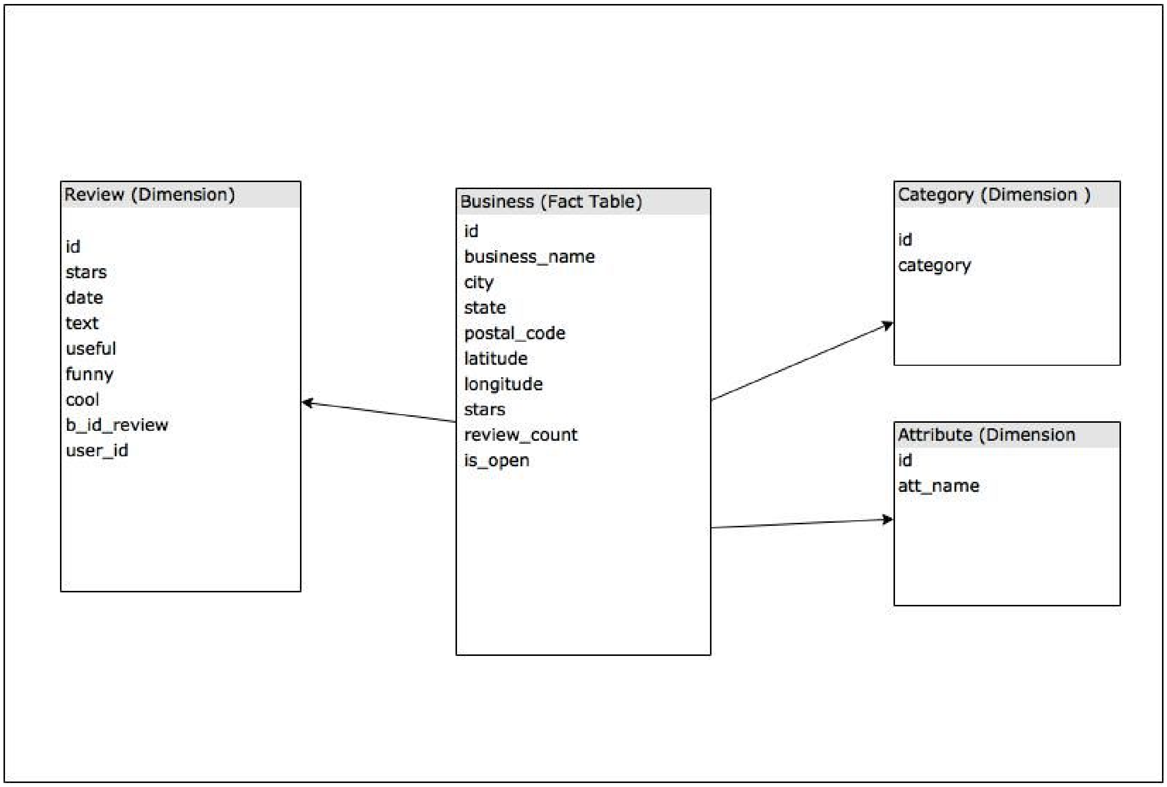
B. <u>Building the Data Warehouse</u>

In order to efficiently implement data mining process we constructed a data warehouse (for the reasons mentioned in Preprocessing Data Number 8) using one of the Google services BigQuery (Google BigQuery Doc). BigQuery is a fully managed, extremely fast, cloud based analytical query service for massive datasets which enables real time analysis.

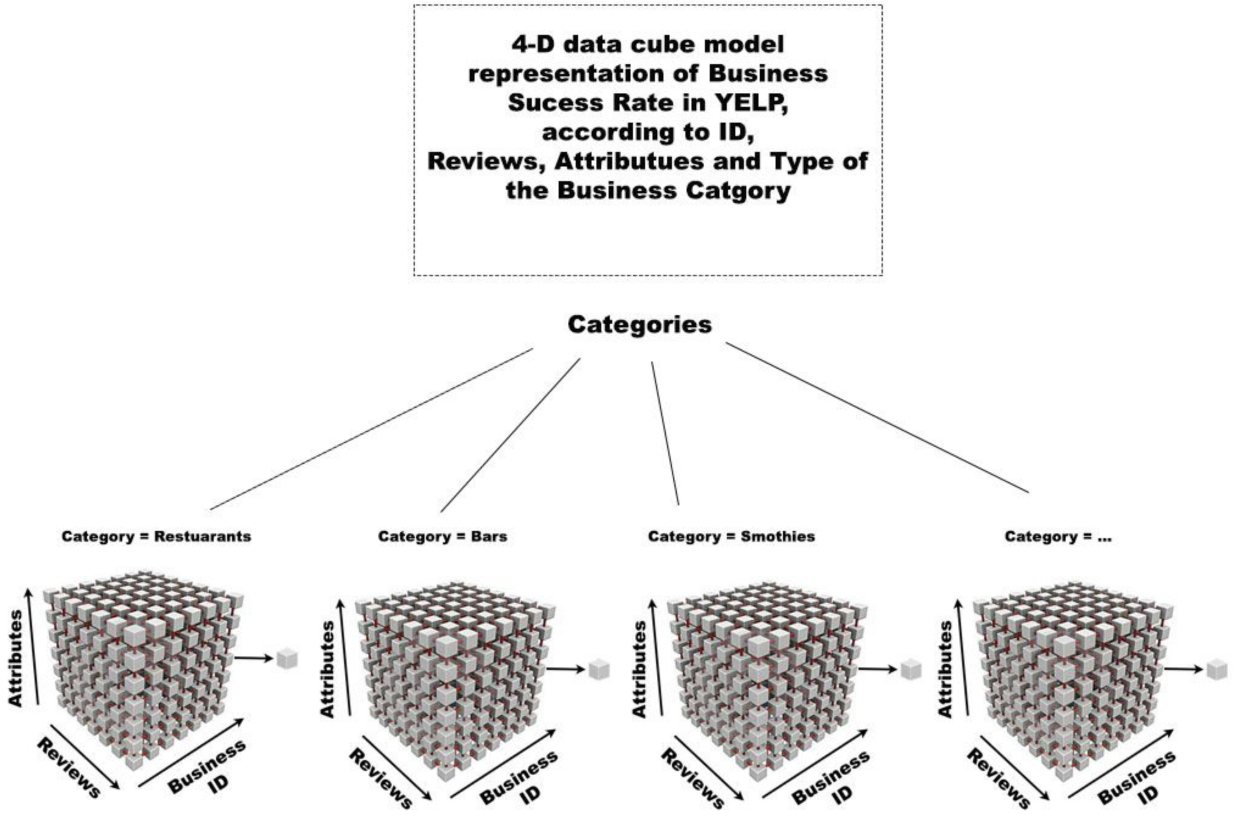
Resulted warehouse consisted of:

- 1 Fact Table:
  - Business
- 3 Dimensions:
  - Attributes
  - Reviews
  - Categories

Schema :



Further, migrating data to BigQuery led to implementation of a 4D Data Cube:


4-D data cube model representation of Business Sucess Rate in YELP, according to ID, Reviews, Attributues and Type of the Business Catgory
Categories
Category = Restuarants
Category = Bars
Category = Smothies
Category = ...
Attributes
Reviews
Business ID

C. Data Mining Process

We performed our algorithms in R and Python.  Types of algorithms that we believed would provide substantial results are:

- Market Basket Analysis
- K-means Algorithm
- Random Forest
- Naive Bayes Classification
- Neural Network
- Sentiment Analysis

## III. Implementation Issues and Solutions

The Yelp dataset was very large, and this caused us issues from the start. We tried several ways to run the .sql file and most of them failed due to the size. This caused us to have to use cloud services, but only after many failed attempts at different local machine solutions. The first cloud service we tried, AWS, was very expensive and also quite difficult to learn and navigate. Eventually we discovered Google Cloud Platform, and one of us hosted the SQL database on there while the rest of us were able to connect remotely.

Once we had access to the database, we began the process of cleaning the data. The fact that this data is very messy and full of inconsistencies was a tremendous challenge for us. Ultimately we were not able to fully clean and verify all of the data, however, we did make a lot of progress during the short time that we had. Much of the cleaning involved simply removing businesses that were of the wrong category or wrong location. Thanks to the abundance of data, this was not an issue. Certain attributes, like is_open, we had to rely upon for accuracy, as we had no way to verify 41,000 businesses in the short time that we had, however, we discovered some inaccuracies in that as well.

Once preprocessing was done, and we moved on to implementing algorithms, we still had issues with the RAM on our local machines. Trying to do stepwise feature selection, Apriori, or Random Forest was impossible on the full data set. Originally we got around this problem by using random sampling of the data, but we soon realized that without aggregation and denormalization we had a tremendous amount of redundancy in the dataset. Although we had 41,000 businesses, we were dealing with over 1.4 billion rows after all the joins it took to get the information we wanted. To deal with this problem we decided to aggregate the dataset (James 201) so that one row of the data would be all the information for 1 business. (Refer to the preprocessing section for details)

Despite aggregation giving us only 41,000 rows, some of the algorithms were still requiring too much memory to run the whole set on our local machines. We simply could not create a tibble or a dataframe from the whole business data and then use some of the algorithms like stepwise feature selection or Apriori. This is when we learned about more features in cloud services. We created Hadoop Spark clusters in

Google Dataproc (Google DataProc Doc) allowing us to get the computing power and memory required. Once we knew how to use clusters, it was a simple matter to install RStudio server, grab the data from BigQuery, and run any algorithm we wanted on the whole dataset. We also learned how to use Jupyter notebooks on the cluster to run any Python code we wished to implement. This ultimately freed us for anything we wanted to do on the entire dataset.

## IV. Analysis and Results

Market Basket Analysis:

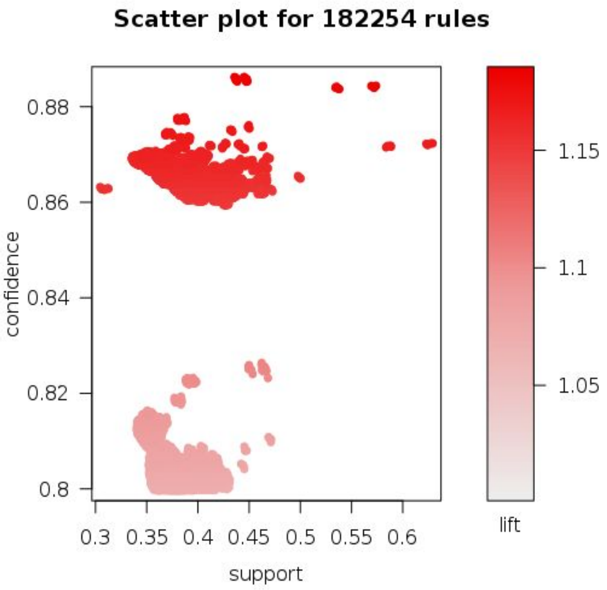
Apriori algorithm was implemented on a set of attributes belonging to various eateries in the US (Hahsler 2). These attributes are one-hot encoded (0-1 encoding), and include features like “Smoking, Catering, GoodForGroups, Music, AcceptsCreditCards, BikeParking” and 26 more. Apriori algorithm was implemented using R package “arules.” Relative support was set at 40% and confidence was set at 75%. Rules are of length 2 or more, with a maximum of length 8. We chose these parameters since we have relatively high number of  sets (>41,000), and because of the relative inefficiency of the algorithm, we ran it in RStudio on a Google Cloud Platform DataProc Cluster.

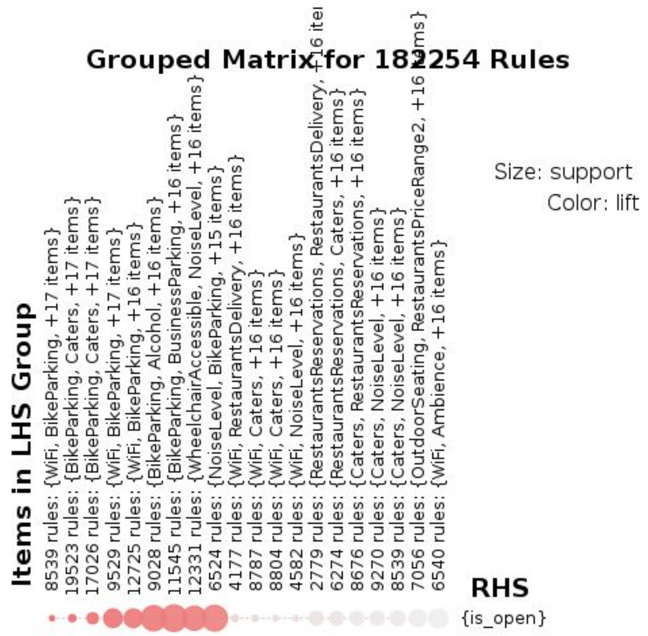
Code Sample in R:

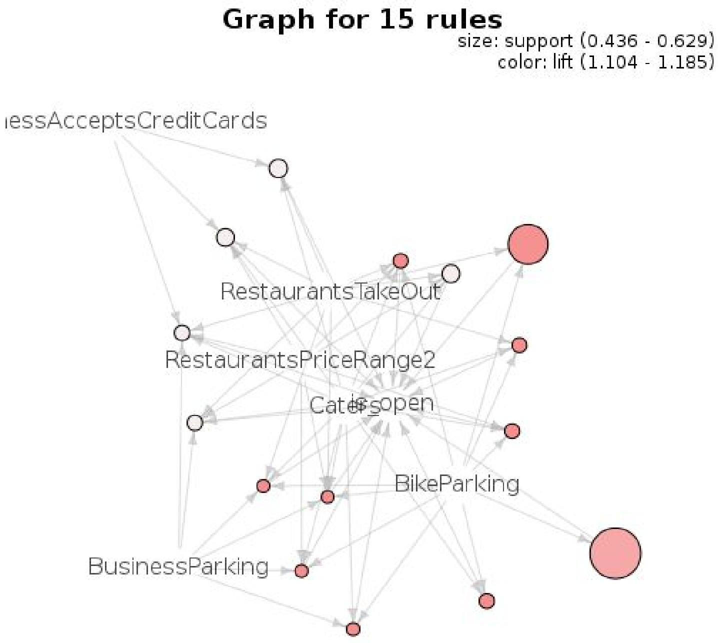
```
>rules<-apriori(att.matrix,parameter=list(minlen=2,supp=.4,conf=.75,maxlen=8),appearance = list(rhs=c("is_open"),default="lhs"),control=list(verbose=F))
```

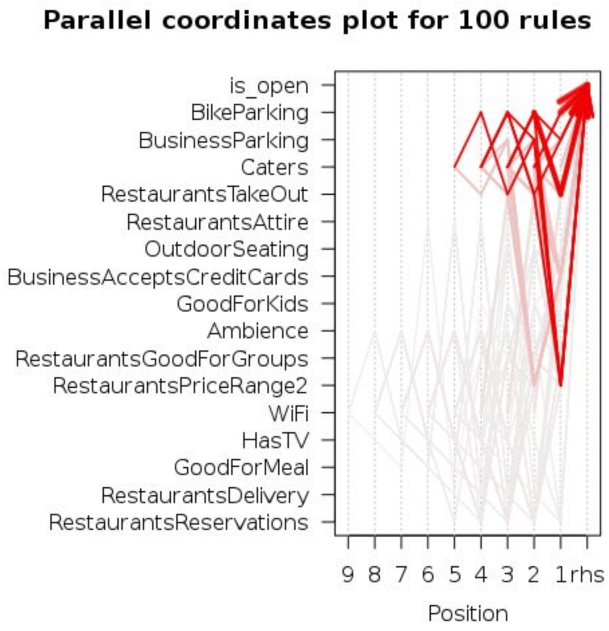
This code focuses the rules on the feature we are trying to predict: the eatery is open (i.e. not out of business). These are top 3 rules that were generated by Apriori Algorithm sorted by lift:

```
    lhs                              rhs       support   confidence   lift   count
[1] {RestaurantsPriceRange2,
     BusinessParking,
     RestaurantsTakeOut,
     BikeParking,
     Caters}                => {is_open} 0.4361684  0.8861191 1.184643 18200
[2] {RestaurantsPriceRange2,
     RestaurantsTakeOut,
     BikeParking,
     Caters}                   => {is_open} 0.4451794  0.8860059 1.184492 18576
[3]  {BusinessParking,
     RestaurantsTakeOut,
     BikeParking,
     Caters}                   => {is_open} 0.4370312  0.8859308 1.184392 18236
```




Grouped Matrix for 182254 Rules
Items in LHS Group
8539 rules: {WiFi, BikeParking, +17 items}
19523 rules: {BikeParking, Caters, +17 items}
17026 rules: {BikeParking, Caters, +17 items}
9529 rules: {WiFi, BikeParking, +17 items}
12725 rules: {WiFi, BikeParking, +16 items}
9028 rules: {BikeParking, Alcohol, +16 items}
11545 rules: {BikeParking, BusinessParking, +16 items}
12331 rules: {WheelchairAccessible, NoiseLevel, +16 items}
6524 rules: {NoiseLevel, BikeParking, +15 items}
4177 rules: {WiFi, RestaurantsDelivery, +16 items}
8787 rules: {WiFi, Caters, +16 items}
8804 rules: {WiFi, Caters, +16 items}
4582 rules: {WiFi, NoiseLevel, +16 items}
2779 rules: {RestaurantsReservations, RestaurantsDelivery, +16 ite
6274 rules: {RestaurantsReservations, Caters, +16 items}
8676 rules: {Caters, RestaurantsReservations, +16 items}
9270 rules: {Caters, NoiseLevel, +16 items}
8539 rules: {Caters, NoiseLevel, +16 items}
7056 rules: {OutdoorSeating, RestaurantsPriceRange2, +16 items}
6540 rules: {WiFi, Ambience, +16 items}
Size: support
Color: lift
RHS
{is_open}


Graph for 15 rules
size: support (0.436 - 0.629)
color: lift (1.104 - 1.185)
ıessAcceptsCreditCards
RestaurantsTakeOut
RestaurantsPriceRange2
Caters
is_open
BikeParking
BusinessParking



The following attributes are highly indicative of a restaurant being in business vs. out-of-business:

- Restaurant price range
- Parking for both cars and bikes
- Take-out
- Catering
- Accepting credit cards

Negative patterns that were found with Apriori:

| **lhs** | **support** | **confidence** | **lift** | **count** |
|---|---|---|---|---|
| {wheelchair_neg,tableservice_neg} => {is_open} | 0.2177008 | 0.816833 | .092016 | 9084 |

Wheelchair accessible and table service are pure negative association rules for the restaurant being open. In other words, if a restaurant does not have table service and does not have wheelchair access, they are open. This is probably owing to the large amount of fast food restaurants in our dataset.

## Association Rules Using K-means

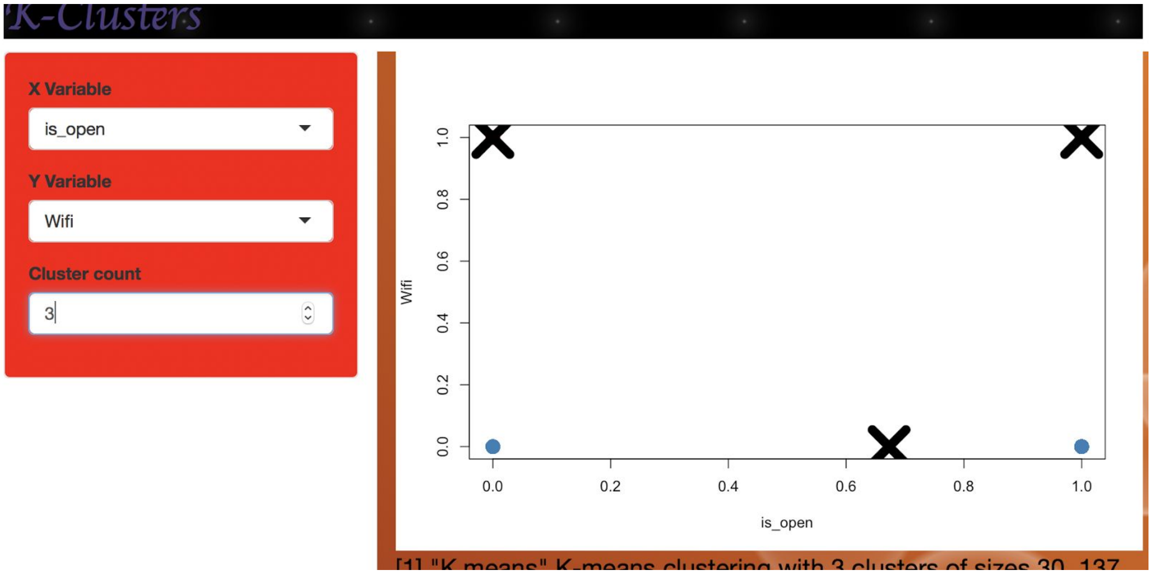
During project progress we decided to perform unsupervised learning on the Yelp Dataset because we want to discover hidden patterns and groupings in data related to restaurants closing. Therefore, we decided to implement K-means (Teetor 240) rather than work on clustering.

We chose to look into restaurants and determine whether or not there is a pattern between restaurants that are (Is-Open-true AND has Wifi-true) and (Is-Open-False AND Has Wifi-false). The reason for the chosen pattern is very simple: as a group we are very tech motivated and we assume that restaurant success can be affected by the presence of the wifi service in their location. Therefore, the proposed hypothesis was:

H: “Restaurants are more likely to close if they do not have Wifi.”

Surprisingly, with three clusters, no matter the fact that a restaurant is opened or closed having Wifi does not determine whether your restaurant will stay open or not. This was also supported previously in our Apriori algorithm.

Conclusion: No relation between Have-Wifi attribute and Is-Open.



K-Clusters

X Variable: is_open

Y Variable: Wifi

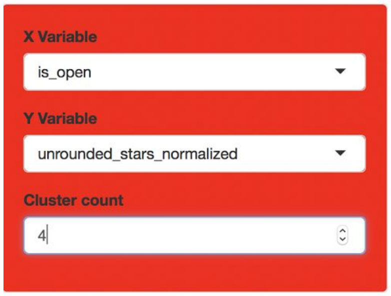
Cluster count: 3

```
[1] "K means" K-means clustering with 3 clusters of sizes 30, 137
```

Another question we decided to ask, is there is a hidden pattern between restaurants that are opened and/or closed and the number of stars that the restaurants has? Are the stars an indicator of the popularity of the restaurant?

Set Hypothesis:

H: “Restaurants with a high ranking tend to stay open. Extreme circumstances they tend to close”



X Variable

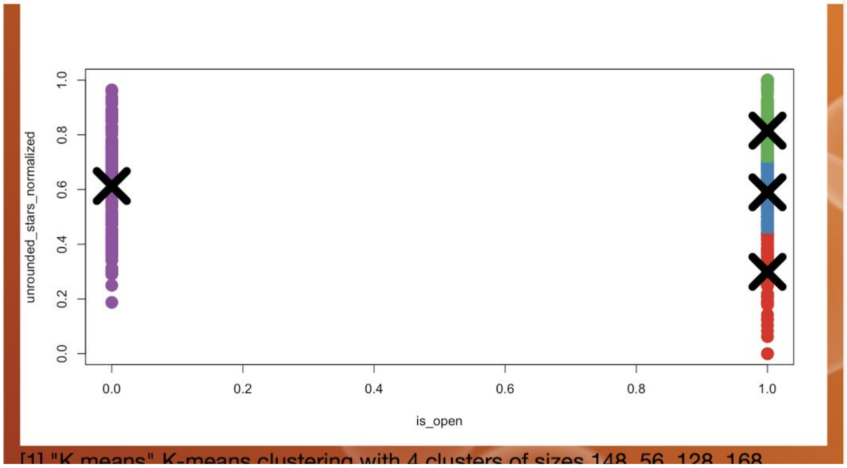
is_open

Y Variable

unrounded_stars_normalized

Cluster count

4



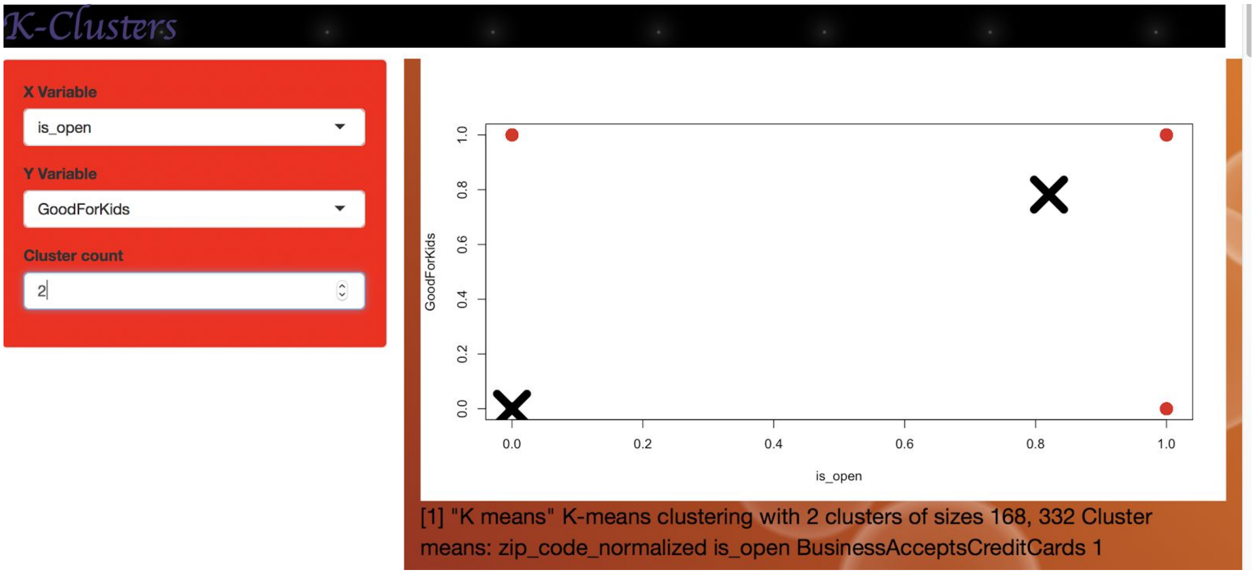
[1] "K means" K-means clustering with 4 clusters of sizes 148, 56, 128, 168

Four clusters show that it is trivial to distinguish a hidden pattern between restaurants that are closed and open. Restaurants that are closed have lower star rating in comparison to restaurants that are opened.

Conclusion:  Is-Open attribute is affected by the Star Ranking.

As a third hypothesis, we decided to determine if “GoodForKids” is an indicator that the restaurant is open or closed.

Hypothesis: “There is a positive relationship between GoodForKids-true attribute Is-Open true and a positive relationship between GoodForKids-false and Is-Open-false.”



K-Clusters

X Variable

is_open

Y Variable

GoodForKids

Cluster count

2

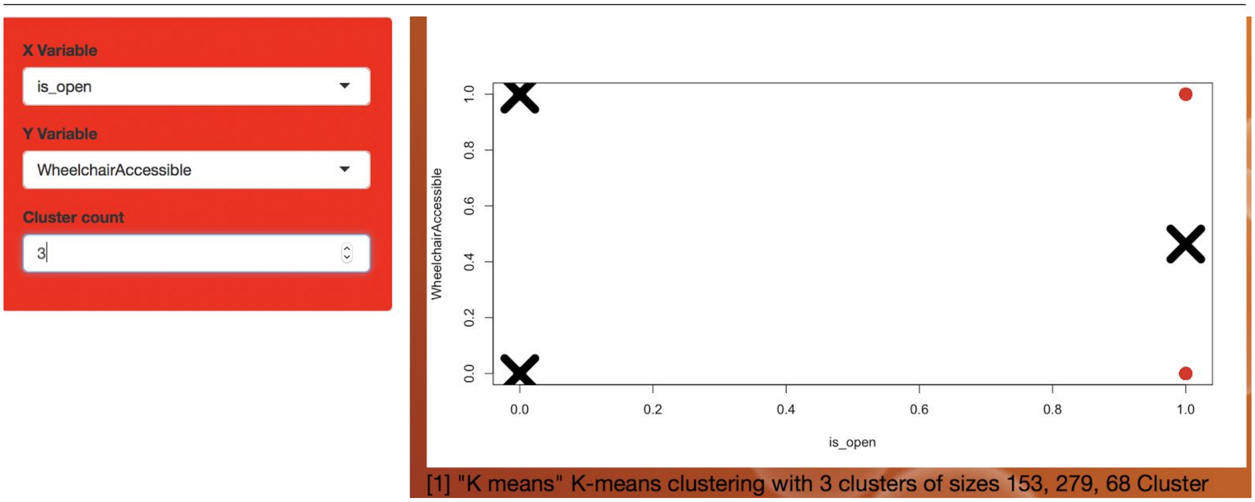
[1] "K means" K-means clustering with 2 clusters of sizes 168, 332 Cluster means: zip_code_normalized is_open BusinessAcceptsCreditCards 1

With two clusters, we can discover that Restaurants that are opened tend more to be good for kids, rather than restaurants that were closed. That could be a factor why certain restaurants closed.

Conclusion: GoodForKids Attribute has a positive impact on Is-Open.

Finally, another hidden pattern we thought would be interesting to look at: does “WheelChairAccessible” influence business success?

Hypothesis: “Existence of WheelChairAccessible attribute will positively influence business success.”



X Variable

is_open

Y Variable

WheelchairAccessible

Cluster count

3

[1] "K means" K-means clustering with 3 clusters of sizes 153, 279, 68 Cluster

With having three clusters, we have discovered that restaurants that tend to be closed are WheelChairAccessible comparing to restaurants that are opened.

Conclusion: WheelChairAccessible attribute negatively influence business success. Again, we saw in Apriori that wheelchair accessibility was a strong factor in business success.

K-means analysis Conclusion:

After analyzing the attributes of the Yelp dataset, we have discovered that certain attributes negatively influence a restaurant being open, while other attributes positively influence a restaurant being open. Additionally, some attributes do not determine if a restaurant will be opened or closed.

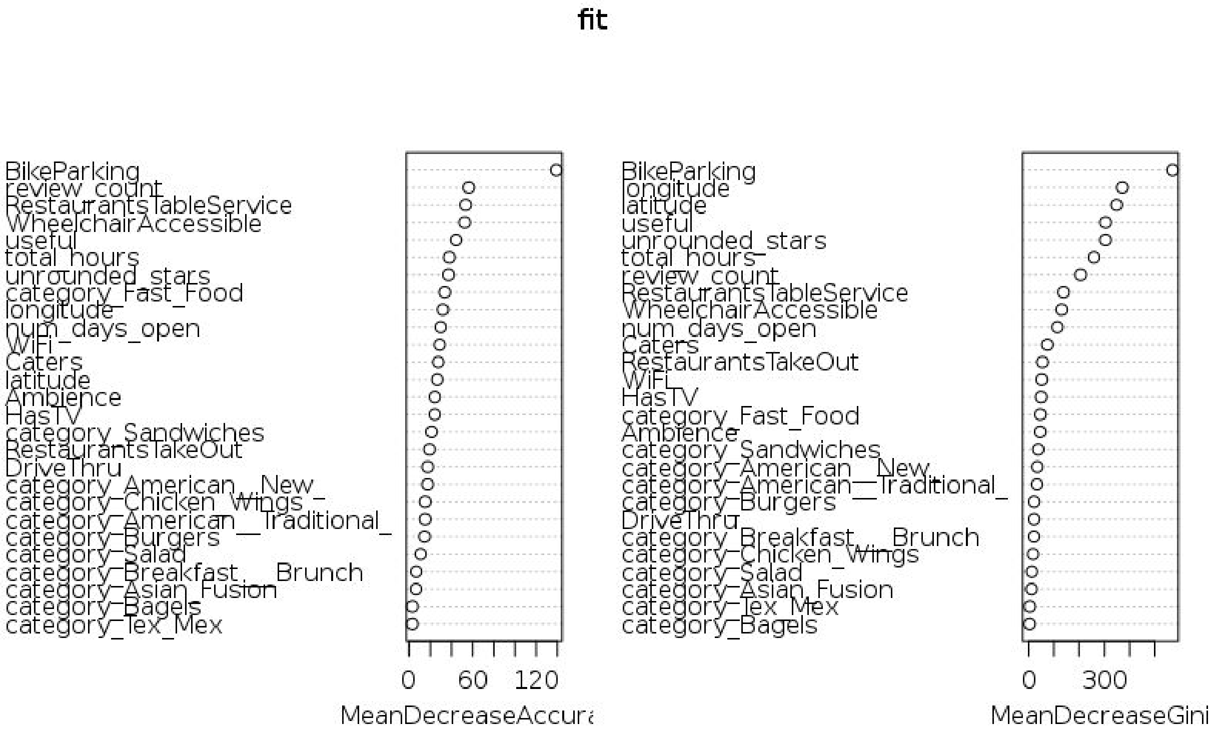
## Random Forest Predictive Model

We wanted to determine a model to predict if a business is open or closed based on the attributes given in the Yelp dataset using as many predictive attributes as possible without introducing redundancy (Hahsler 2). Categories and attributes were both pared down using entropy-based feature selection (James 344) to the top 9 attributes and top 11 categories:

```
is_open ~ BikeParking + Caters + RestaurantsTakeOut + category.Fast.Food +
          WiFi + DriveThru + WheelchairAccessible + RestaurantsTableService +
          HasTV + Ambience + category.Salad + category.Chicken.Wings +
          category.Burgers + category.Sandwiches + category.Breakfast...Brunch +
          category.Tex.Mex + category.American..New. + category.American..Traditional. +
          category.Asian.Fusion + category.Bagels
```

The numeric features were checked for correlation using Pearson correlation coefficient (James 99). Of any group of correlated variables, only one was kept in the data. For example, the variables funny, cool, and useful, all user tags for reviews, were highly correlated with coefficients >.85, therefore only useful was retained and funny and cool were both discarded.

Random Forest (Liaw 4) was run in RStudio on a Google DataProc Cluster using the “randomForest” package. The number of decision trees was 500, the OOB error was 15.86%, and the number of variables tried at each split was 5.



The model was made using all of the above attributes.

Random Forest Conclusion:

As can be seen in the image, some of the most important variables determining business success include Bike Parking, Review Counts, Useful Counts, Star Counts, and Total Hours open per week. These are all in the top 10 regardless of looking at accuracy or Gini. The model’s final accuracy was 84.2%.

## Naive Bayes Classifier

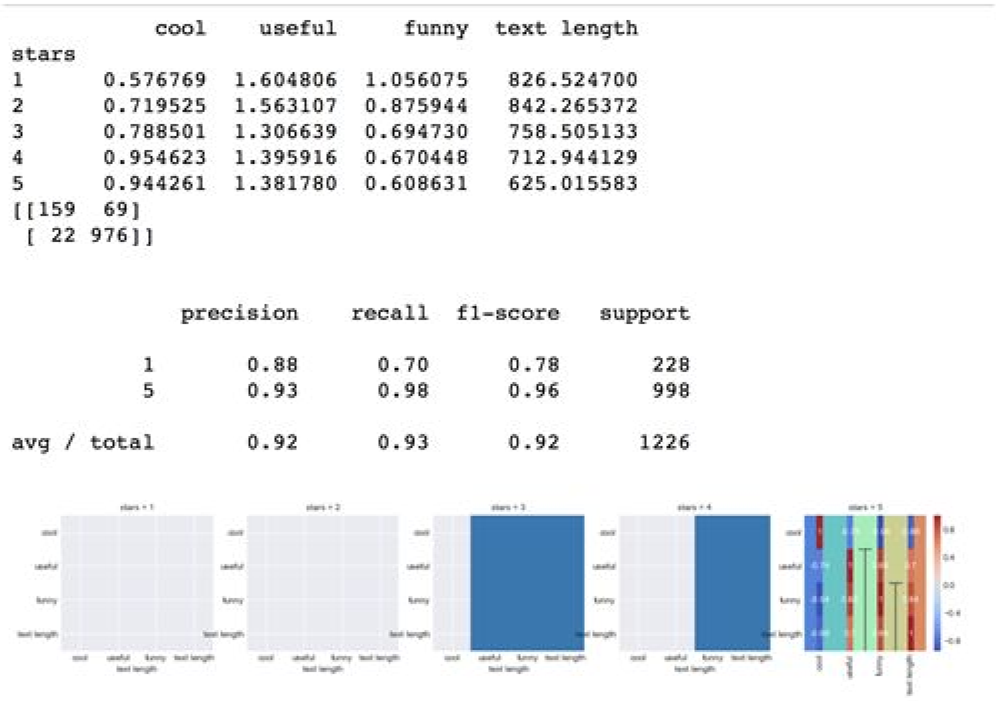
A Naive Bayes is in the family of simple probabilistic classifiers based on Bayes theorem. Naive Bayes is a simple technique for constructing classifiers. An advantage of naive Bayes is that is requires a small amount of training data (Majka 4). This was the perfect technique to apply to our Yelp Data Set.

Based on a Conditional Probability Model, we could ask ourselves: Based on text in a review, can we predict the rating?

We did this with a simple approach.

1. We first took the length of reviews
2. We then took if that review was cool
3. We then took If that review was useful
4. We then took if that review was funny
5. We then took the stars for that review

With Bayes, we were able to predict, based on the length, cool, useful, and funny, what are the stars.



```
           cool    useful     funny  text length
stars
1      0.576769  1.604806  1.056075   826.524700
2      0.719525  1.563107  0.875944   842.265372
3      0.788501  1.306639  0.694730   758.505133
4      0.954623  1.395916  0.670448   712.944129
5      0.944261  1.381780  0.608631   625.015583
[[159  69]
 [ 22 976]]

             precision    recall  f1-score   support

          1       0.88      0.70      0.78       228
          5       0.93      0.98      0.96       998

avg / total       0.92      0.93      0.92      1226
```

After looking at the outcome of Bayes versus the true values, our results are up to par. Usually longer reviews have a lower rating as customers are not satisfied. Looking at the outcome for cool reviews, reviews that have four stars are cooler than reviews that have five stars. After looking at the reviews that are useful, it is reasonable that 1 star has the highest review score. People like to know when not go to to a restaurant. Finally, it makes since that one star has the highest review rating when it is labeled as funny. As users get a kick out of the venting people do regarding restaurants that gave them poor experiences.

Bayes Classifier Conclusions:

1. Length of review matters if a restaurant gets high stars

2. Businesses that have low stars  usually are rating in high in being useful and funny

## Neural Network

Looking at the reviews and text to get an understanding of a review being a factor in a restaurant being open or closed, a neural network was built to determine if stars are a factor in a restaurant being open or closed.

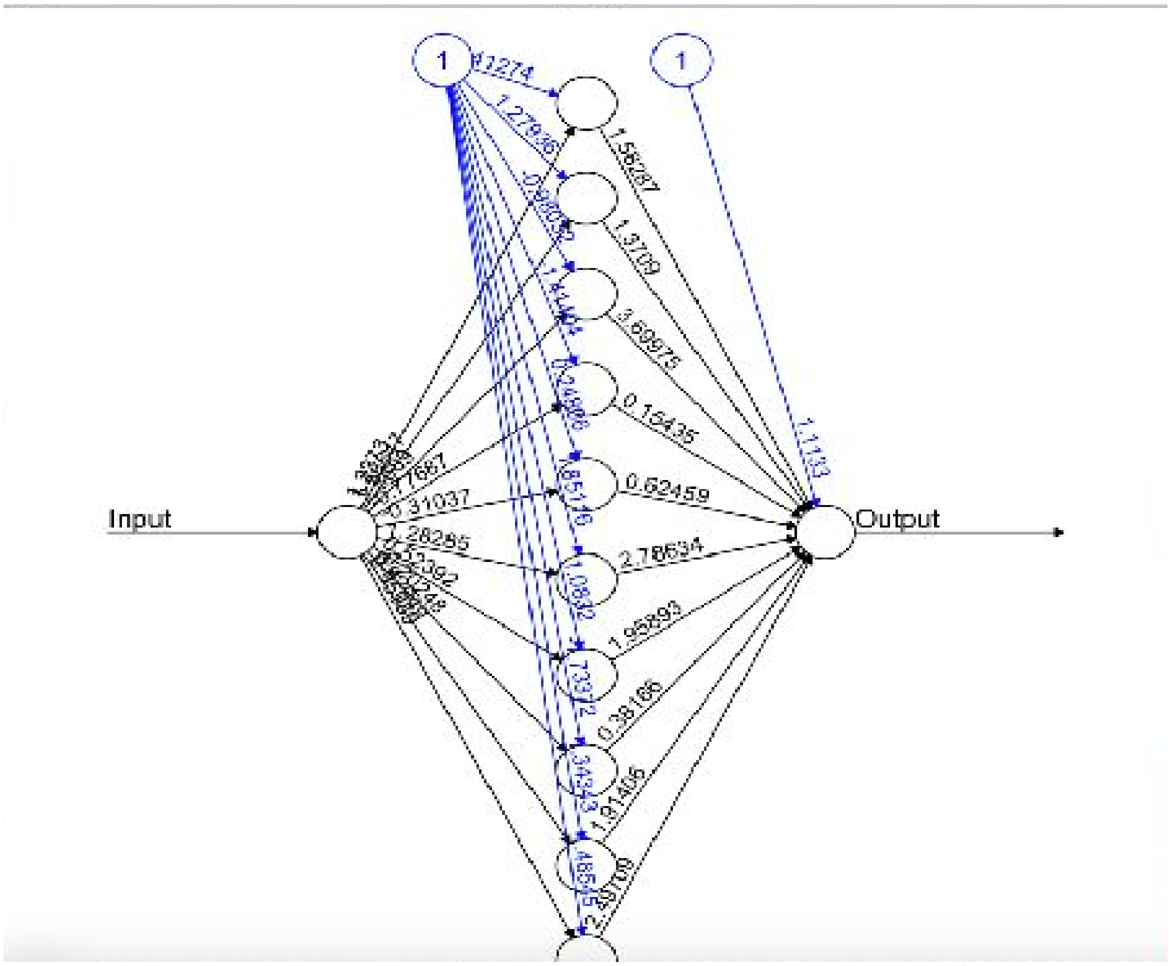
In R, we created a neural network with the “neuralnet” library in R (Fritsch 5). This is a basic neural network, as our perceptron has one input. The input is the number of stars per restaurant. With our activation function we are taking into consideration reviews that are useful, cool, and funny, even though they are correlated.

Our model has three hidden layers. The black lines show the connections with the weights. The weights are calculated using the backpropagation algorithm. The backpropagation algorithm is one of the typical algorithms used in artificial neural networks to calculate the error contribution of each neuron after a batch of data is processed. The blue lines explain the bias term. The predicted stars have an error of 0.077. We have around 146 steps. We trained the neural network to have at least 10 hidden errors. For stopping criteria for this neural network, we used the derivatives of the errors (Fritsch 6).

Our neural network gave us a valuable output that the stars for restaurant reviews seem pretty accurate, as it is similar to the input we gave the network. To make sure, we tested our output with a sample training set to see if we got a better version of our results. Our results were around the same, which tells us the stars given to a restaurant are pretty accurate. This tells us the stars will contribute to a success of a business - if it will stay opened or closed. Even though we fed into this neural network if the review was useful, cool, or funny, they are correlated.

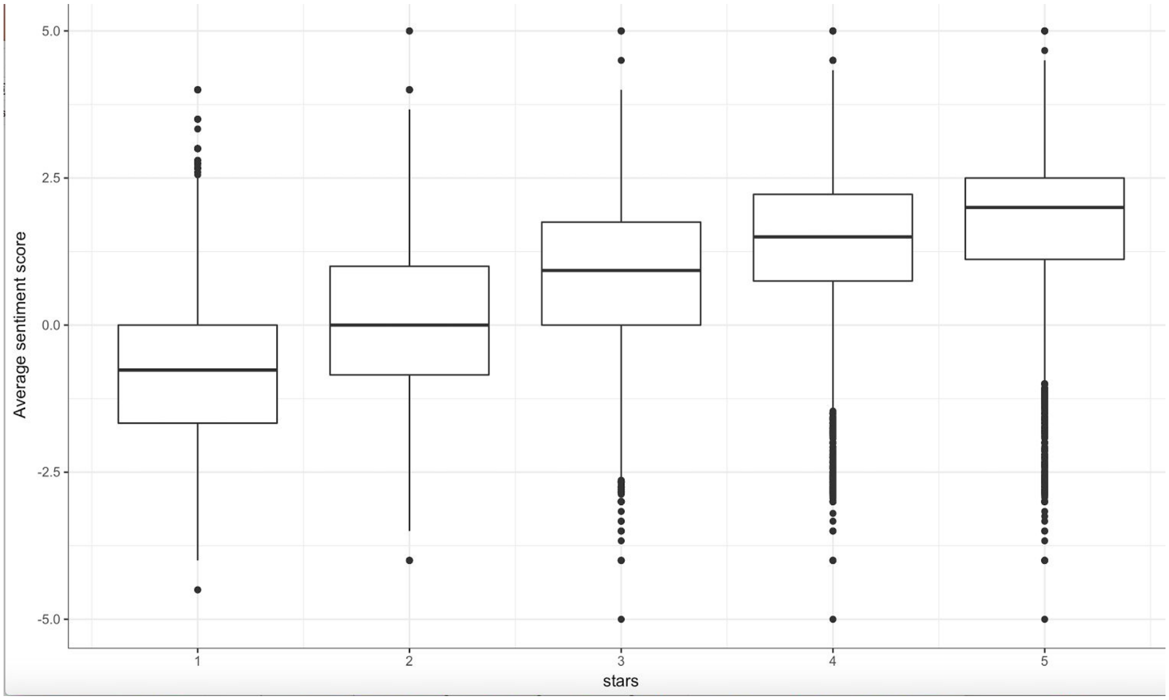
Neural Network Conclusion:

Stars are the most important factor for the business. The stars will determine the future of the business.



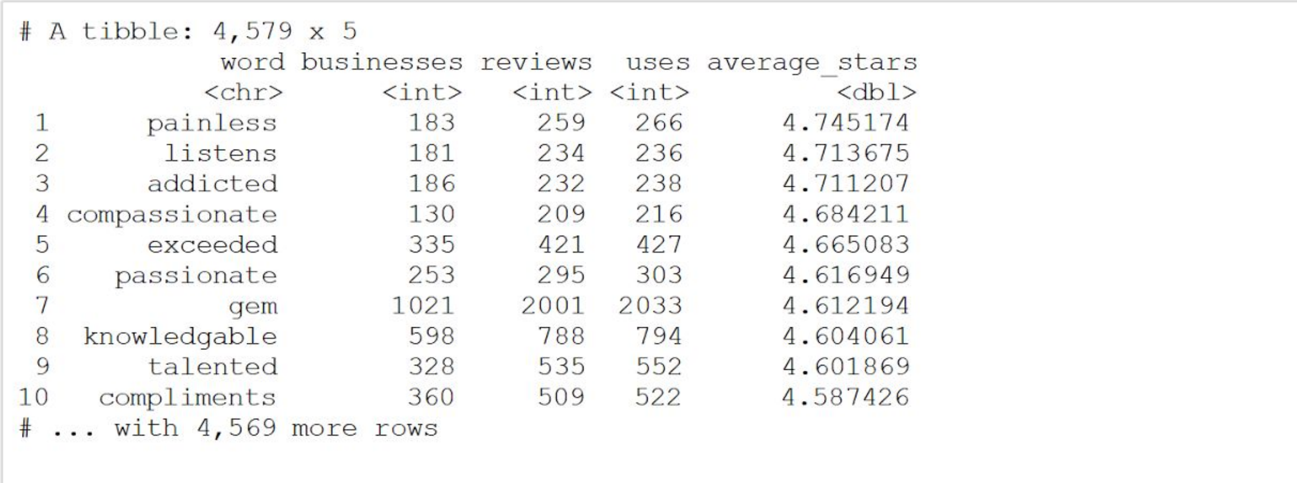
## Sentiment Analysis

We have a large amount of reviews in the Yelp dataset, and we wanted to include them in our data mining because of their potential for huge impact on the business. Reviews left by customers can determine what the future holds for a business, as well as what a business needs to focus on to boost success or avoid closing. To begin analysis of the text reviews, we started with sentiment analysis (Feuerriegel 5) of words within the text.

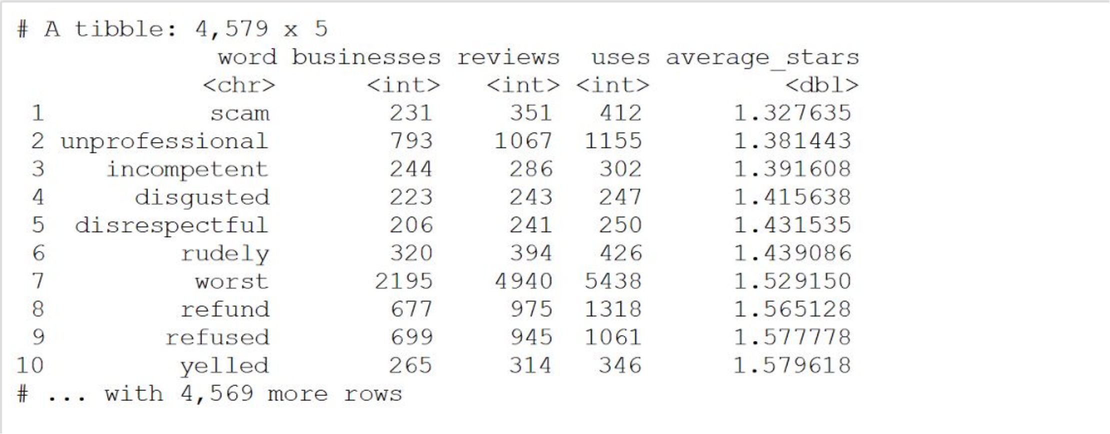


Boxplot for Average Sentiment Score by Stars

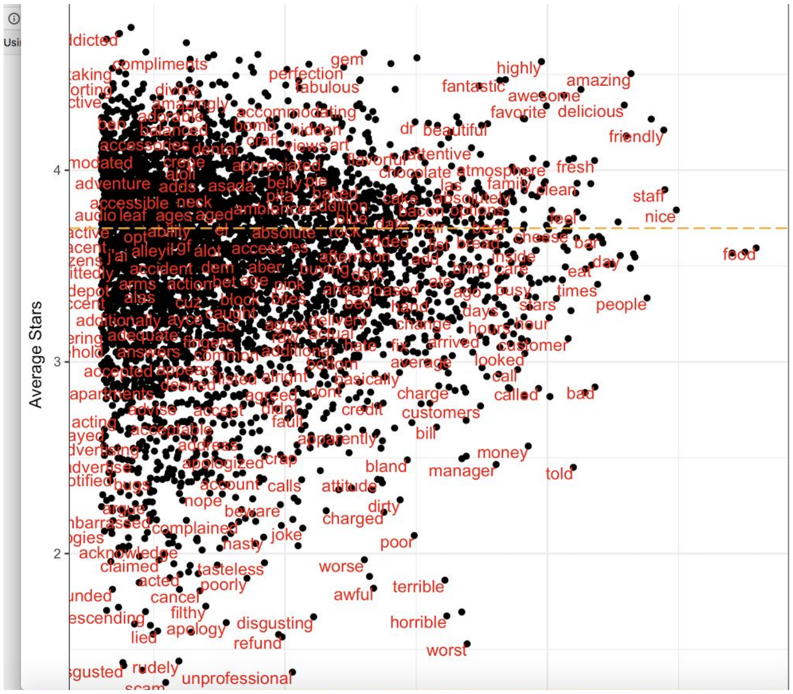
We can tell is that our sentiment score is correlated with positive ratings, despite having a large prediction error.



```
# A tibble: 4,579 x 5
            word businesses reviews  uses average_stars
           <chr>      <int>   <int> <int>         <dbl>
 1      painless        183     259   266      4.745174
 2       listens        181     234   236      4.713675
 3      addicted        186     232   238      4.711207
 4 compassionate        130     209   216      4.684211
 5      exceeded        335     421   427      4.665083
 6    passionate        253     295   303      4.616949
 7           gem       1021    2001  2033      4.612194
 8  knowledgable        598     788   794      4.604061
 9      talented        328     535   552      4.601869
10    compliments       360     509   522      4.587426
# ... with 4,569 more rows
```

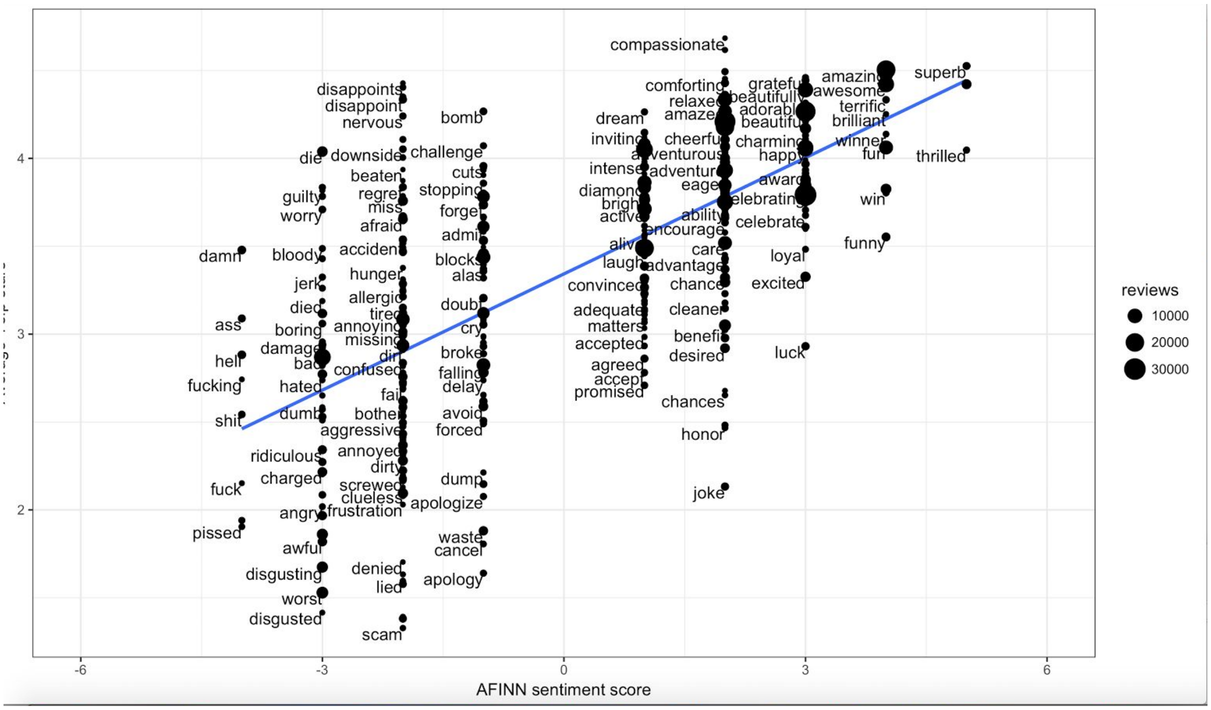


```
# A tibble: 4,579 x 5
            word businesses reviews  uses average_stars
           <chr>      <int>   <int> <int>         <dbl>
 1          scam        231     351   412      1.327635
 2 unprofessional       793    1067  1155      1.381443
 3    incompetent       244     286   302      1.391608
 4     disgusted        223     243   247      1.415638
 5  disrespectful       206     241   250      1.431535
 6        rudely        320     394   426      1.439086
 7         worst       2195    4940  5438      1.529150
 8        refund        677     975  1318      1.565128
 9       refused        699     945  1061      1.577778
10        yelled        265     314   346      1.579618
# ... with 4,569 more rows
```



Plot of Positivity

Which words broke the trend of being a positive and or negative review?



Sentiment Analysis Conclusion:

The positive and negative words in a review indicate what the star count of the restaurant will be, and the star count, as known from previous mining methods, is highly indicative of a restaurant's success.

## V. Summary

In mining the Yelp dataset for the 10th annual Yelp Challenge, our group discovered a lot about what makes restaurants successful, and this information can prove valuable to businesses looking to improve their restaurant's profit and success. Apriori itemset mining showed us that a combination of reasonable pricing, bike and automobile parking, catering, and take-out is a winning combination for a restaurant. It also shows the negative associations of no wheelchair access and no table service together being associated with a restaurant being open, which is probably due to fast food restaurants with drive-thru windows. Random Forest showed us that bike parking is surprisingly very important for a business, people are more inclined to go to a restaurant that has explicit parking for their bicycles. We can also see that total hours open, total days of week open, star counts, review counts, and useful tags on reviews are highly indicative of a place’s success. Businesses that might feel that they should provide WiFi for their customers probably should not bother, because our mining through k-means showed that this is probably not a big factor to their success, however, being more kid friendly should be considered, and efforts should be made to keep their star ranking up. One way to do that, according to our sentiment analysis, is to watch their reviews. We found that positive and negative words are definitely correlated to the business’s star count. The more positive words in a review, the higher the star count. Interestly, businesses with perfect stars (5) should probably not get too comfortable, as our mining also showed that having 4 stars is more indicative of the business staying open that 5 stars. Also, we conclude that length of review matters. People who are unhappy with their service tend to give longer reviews, and vice versa. Negative reviews got the most funny and useful tags, so businesses should watch out for the supposedly positive funny and useful markings, as they might be indicating the opposite of what they seem.

# Works Cited

Han, Jiawei et al., *Data Mining Concepts and Techniques, Third Edition.* Morgan Kaufman Publishers. 2012.

Hahsler, Michael et al., *Mining Association Rules and Frequent Itemsets.* Version 1.5-4. October 2012. www.http://s2.smu.edu/IDA/arules/ Accessed Oct 2017.

Fritsch, Stefan et al., *Training of Neural Networks*. Version 1.33. August 2016. www.cran.r-project.org/web/packages/neuralnet/neuralnet.pdf Accessed Dec 2017.

Majka, Michael. *High Performance Implementation of the Naive Bayes Algorithm.* Jan 2017. www.cran.r-project.org/web/packages/naivebayes/naivebayes.pdf. Accessed Dec 2017.

Liaw, Andy and Matthew Wiener. *Breiman and Cutler's Random Forests for Classification and Regression.* Oct 2015. www.cran.r-project.org/web/packages/randomForest/randomForest.pdf Accessed Nov 2017

Feuerriegel, Stefan and Nicolas Proellochs. *Dictionary-Based Sentiment Analysis.* Nov 2017. www.github.com/sfeuerriegel/SentimentAnalysis. Accessed Dec 2017.

James, Gareth et al., *An Introduction to Statistical Learning with Applications in R.* Springer Publishing. New York. 2015.

Teetor, Paul. *R Cookbook.* O'Reilly Publishing. Cambridge. March 2011.

De Jonge, Edwin and Mark van der Lon. *An Introduction to Data Cleaning with R.* Statistics Netherlands. The Hauge/Harleen. 2013.

Google Cloud SQL Documentation. www.cloud.google.com/sql/docs/. Accessed Sept 2017.

Google BigQuery Documentation. www.cloud.google.com/bigquery/docs. Accessed Sep 2017.

Google DataProc Documentation. www.cloud.google.com/bigquery/docs. Accessed Nov 2017.

**references (authors, title, publishing source data, date of publication, URL) and you should quote each reference in your report text**

**appendix containing a set of supporting material such as examples, sample demo sessions, and any information that reflects your effort regarding the project**