Predicting rating of restaurants in California, Nevada, Utah and Arizona

Cristián Vildósola Michell

June 2020

Contents

1 Introduction

1.1 Background

There are thousands of restaurants across the United States, if we observe the industry of dinning out, we can realize that total number of restaurants in the United States in 2018 were 660.755, and it keep increasing at 1-2% rate, also "The National Restaurant Association projects overall industry sales will hit a high of $863 billion in 2019, up 3.6% from last year." Based in the channel CNBC, the number of restaurants that fails each year its about 60%, and nearly 80% close before their 5 year anniversary. Also, in this news channel, the most common cause of failure its location and user reactions in social platforms.

1.2 Problem

In this project we are going to use machine learning algorithms with the intention of predict the rating of a restaurant based in some variables we are going to get from Yelp. The goal of this, its identifies where its more possible a business to succeed or if its more important the reviews that people make in order to evaluate the food and the place.

2 Methodology

2.1 Data Adquisition

In order to obtain the data, we used the API offered by Yelp, specifically the business search. Then we proceed to make a few requests and pass the data to pandas data frame. The fields acquired where: business_id, name, address, city, state, postal_code, latitude, longitude, stars, review_count, open, categories of food and a series of attributes. We are going to mainly use the locations coordinates, state, review_count and price. The specific data request were restaurants from California, Nevada, Utah and Arizona. In total we got 1800 places to explore.

The main fields selected for our exploration are: name, longitude, latitude, price, reviews and rating. This variable can have the following values:

1. name: It‘s the name of the city in string type.
2. longitude: It‘s the coordinate of the city in the Y - axis, in float type.
3. latitude: It‘s the coordinate of the city in the Y - axis, in float type.
4. rating: It‘s the users evaluation of a restaurant, it‘s an int value between 0 and 5.
5. price: It‘s the price rate of a restaurant, it‘s an int value between 0 and 3.
6. reviews: It‘s the amount of reviews a restaurant has, it can be any int value.

The data fields are not enough to determine the potential of a prediction or its possible use for modelling out problem, so in the next section we are going to make an exploratory analysis.

2.2 Exploratory Analysis

For our study case we are going to compare 4 states of the United States, these are California, Nevada, Utah and Arizona. In the next plot you can check the distribution of ranking for each one.

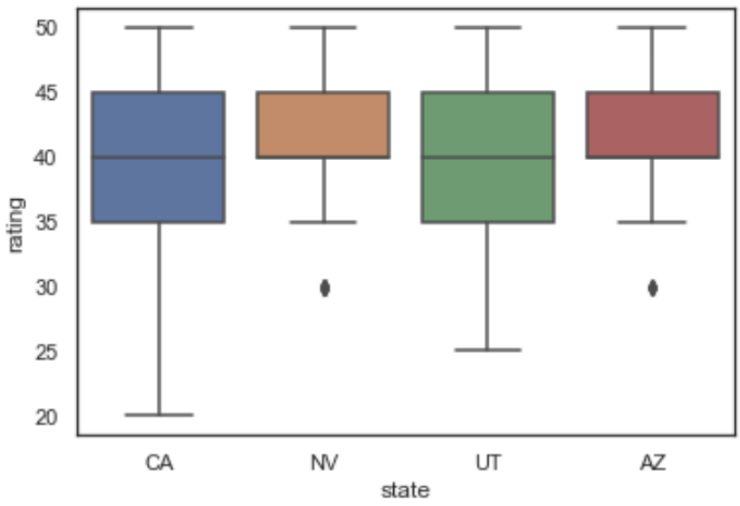

Figure 1: Boxplot of rating by state

As the figure shows, we can see that most of the restourants rating is 40, for example in California we have 25% of the ratings between 35 and 45. Other good example of the rating distribution could be Nevada where 25% of the data has a rating equals to 40.

In the next plot we are going to see the correlation between the data.

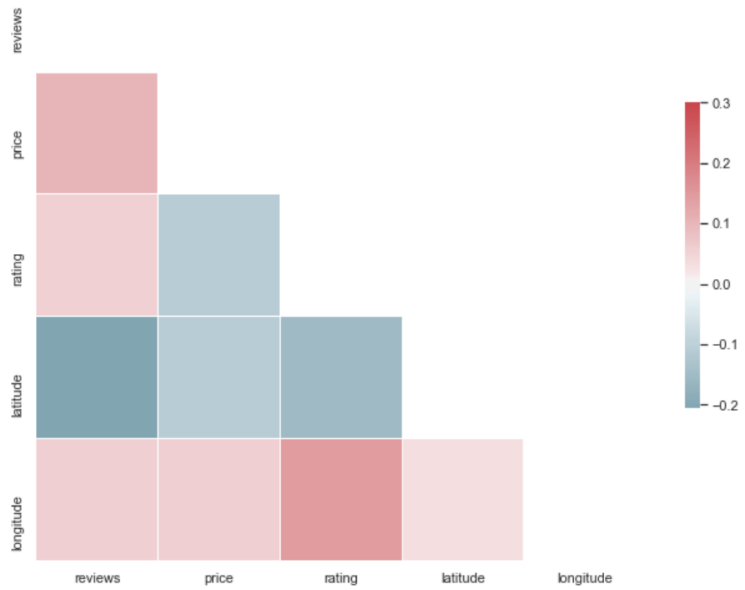

Figure 2: Correlation of variables

The correlation of the data that show our plot, says that most of the variables don‘t relate to each other. By this point we know that the prediction we can make are not going to be easy, because we have

very similar data and it doesn't even have good relationships within itself. In the next analysis we are going to see how price, rating and reviews shows in a scatter plot.

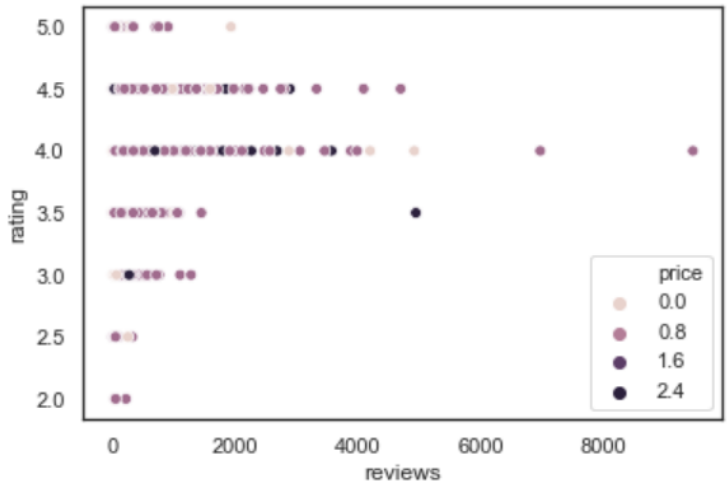

Figure 3: Scatterplot of variables

The last figure is how the restaurants distributes in each state, for this we used the folium package along with python. This plot will give us an idea of the location of each business.

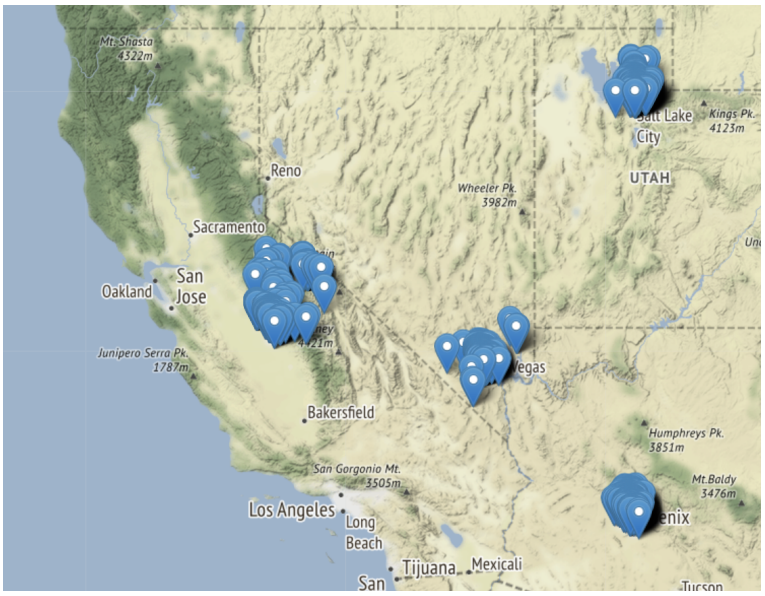

Figure 4: Location of restaurants

The conclusion of our exploratory analysis, its that data seems to give us just a few information for a possible prediction, because of this, we are going to separate the data for each city and check if exists a best approach of predicting the rating in this way.

2.3 Clustering the cities

In the first approach of the clustering we tried to create cluster within all the data, for this task we used the variables: city, price, reviews and rating. For this clustering we decided that latitude and longitude would not give as valuable information because we dont want to create the clusters based in the locations of the restaurants, at least not by this point. The colors of each cluster are represented by, red, blue green and pink for respectively 0, 1, 2 and 3.

This cluster method were called state clustering and its shown in the plot below.

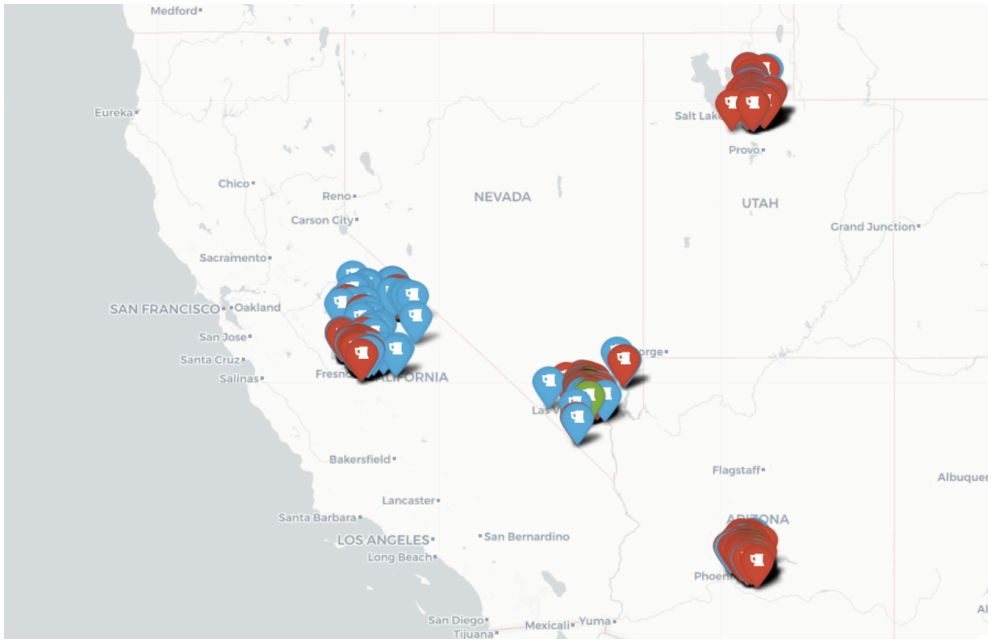

Figure 5: State clustering

In the following charts, we can see how this cluster groups the data for the remaining variables, price, rating and reviews. Also we have a plot including the geo-refence of the data.

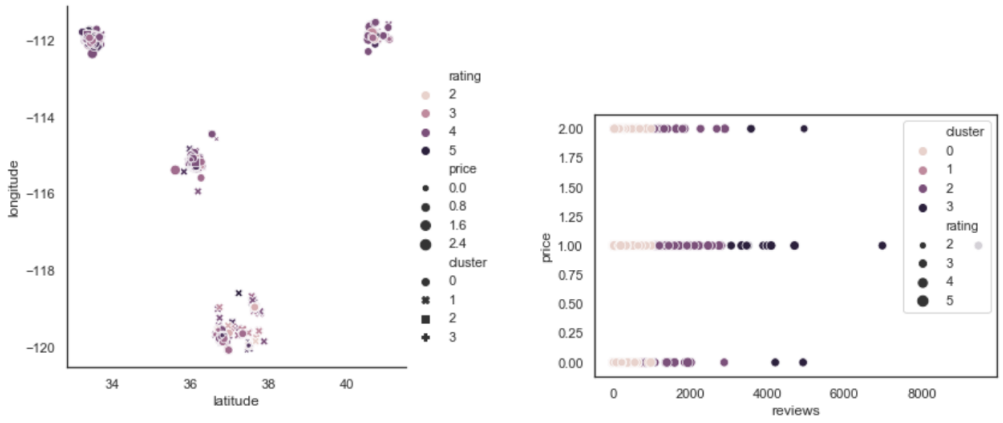

Figure 6: State clustering

In the below chart, we can see the average price and rating and the amount of reviews for each cluster.

	cluster	price	rating	reviews
0	0	0.845347	4.007208	214742.0
1	1	0.952924	4.097004	209751.0
2	2	1.012739	4.121019	217698.0
3	3	1.000000	4.066667	67642.0

Figure 7: State clustering

The cluster 3 is represented by color pink and there are very few restaurants corresponding to it (about 5%). From this analysis, we cant conclude much, so we are going to try the clustering for only the state of California.

2.4 City clustering

For our second clustering attempt, we choose the variables: price, rating, reviews, longitude and latitude, in this case we believe that including location information within the city can reflect some effects of neighbours and luxury locations. For this model we only created this cluster using blue, red and green to represents the cluster 1, 2 and 3 respectively.

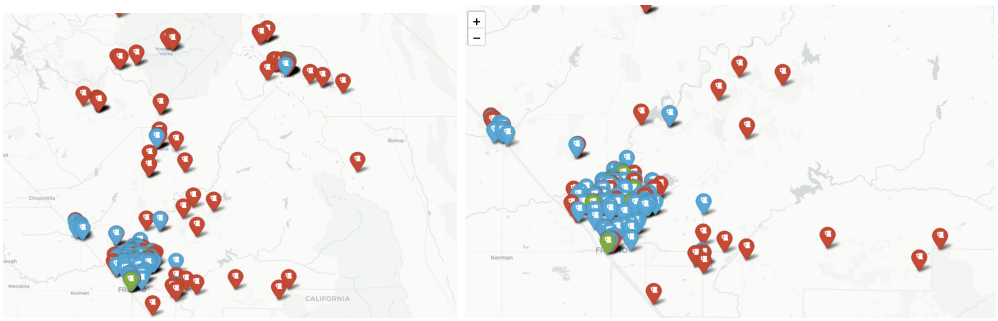

Figure 8: State clustering

Most of the blue cluster are in the inside of the city of Fresno, we continue to check how the clusters are distributed, for this we start analysing the cluster. The Green cluster has the most reviewed restaurants in the city, the blue cluster represents restaurants with a lot of reviews and the red ones are the lesser reviewed restaurants. The classification had sense and represents the distribution of the data.

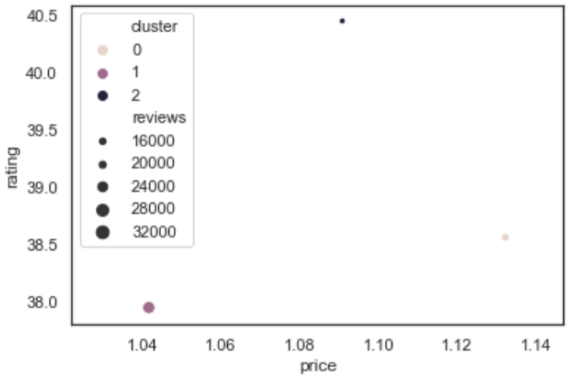

Figure 9: State clustering

2.5 Predictive Modelling

For the predictive modelling we are going to test a multiple linear regression for each city and then we will split each city data and try to predict the possible ranking of the each business in the test group, for this we are going to use the classification method Support Vector Machine (SVM), once this is done we are going to compare the results and see how to improve this predictions.

1. **California:**
   For the multiple regression in California we got the next parameters: the Coefficients were: 2.61475696e+00, 1.02086707e-03, -6.62078710e+00 and 5.03700414e+00, also the Residual sum of squares was 30.17 and the Variance score was 0.03. The SVM prediction had a precision of 0.42 a recall of 0.96 and a f1-score of 0.59. Finally we had an accuracy of 42%.

2. **Nevada:**
   For the multiple regression in Nevada we got the next parameters: the Coefficients were: -2.11281351e-02, -4.49662308e-05, -4.29118440e+00 and 3.84329450e+00, also the Residual sum of squares was 20.44 and the Variance score was -0.03. The SVM prediction had a precision of 0.49 a recall of 1.00 and a f1-score of 0.66. Finally we had an accuracy of 49%.

3. **Utah:**
   For the multiple regression in Utah we got the next parameters: the Coefficients were: -1.43351027e+00, -3.87867851e-04, -2.50725664e+00 and 3.74185296e+00, also the Residual sum of squares was 11.77 and the Variance score was 0.02. The SVM prediction had a precision of 0.39 a recall of 1.00 and a f1-score of 0.56. Finally we had an accuracy of 39%.

4. **Arizona:**
   For the multiple regression in Arizona we got the next parameters: the Coefficients were: -7.31261473e-01, -1.50879128e-03, 3.76237395e+00 and 5.95098308e+00, also the Residual sum of squares was 12.59 and the Variance score was -0.04. The SVM prediction had a precision of 0.44 a recall of 1.00 and a f1-score of 0.61. Finally we had an accuracy of 44%.

2.6 Conclusions

From the experiment we can visualize a good clustering classification of the users, in which we can get the best restaurant by price in the state clustering, by the other side the city clustering allow us to get the restaurants by the amount of reviews and the rating. The downside of the analysis is in terms of predicting the rating of each place. The model didn't perform well, we knew this when we didn't recognize cor relationships between the variables, also the number of data for each variable make the model more complex to recognize the any pattern within the data.
The next steps, could be take more data and try to separate it in more equal sized partitions to test the variables, another point to upgrade is maybe choose more related to the problem and try bigger data sets.

3 Reference

1. **CNBC - Why restaurants fails?**

   https://www.cnbc.com/2016/01/20/heres-the-real-reason-why-most-restaurants-fail.html

2. **CNBC - Budget towars eating out**

   https://www.cnbc.com/2019/08/19/americans-putting-more-of-their-budget-toward-eating-out.html

3. **Data Acquisition**

   https://www.yelp.com/dataset/

4. **Troubleshooting**

   https://github.com/Yelp/yelp-fusion/issues/307

3.1 Data Cleaning

The yelp API allow us the following information for each business:

	Name	Type	Description
0	categories	object[]	A list of category title and alias pairs assoc...
1	categories[x].alias	string	Alias of a category, when searching for busine...
2	categories[x].title	string	Title of a category for display purpose.
3	coordinates	object	The coordinates of this business.
4	coordinates.latitude	decimal	The latitude of this business.
5	coordinates.longitude	decimal	The longitude of this business.
6	display_phone	string	Phone number of the business formatted nicely ...
7	hours	object[]	Opening hours of the business.
8	hours[x].is_open_now	boolean	Whether the business is currently open or not.
9	hours[x].hours_type	string	The type of the opening hours information. Rig...
10	hours[x].open	object[]	The detailed opening hours of each day in a week.
11	hours[x].open[x].day	int	From 0 to 6, representing day of the week from...
12	hours[x].open[x].start	string	Start of the opening hours in a day, in 24-hou...
13	hours[x].open[x].end	string	End of the opening hours in a day, in 24-hour ...
14	hours[x].open[x].is_overnight	boolean	Whether the business opens overnight or not. W...
15	id	string	Unique Yelp ID of this business. Example: '4kM...
16	alias	string	Unique Yelp alias of this business. Can contai...
17	image_url	string	URL of photo for this business.
18	is_claimed	bool	Whether business has been claimed by a busines...
19	is_closed	bool	Whether business has been (permanently) closed
20	location	object	The location of this business, including addre...
21	location.address1	string	Street address of this business.
22	location.address2	string	Street address of this business, continued.
23	location.address3	string	Street address of this business, continued.
24	location.city	string	City of this business.
25	location.country	string	ISO 3166-1 alpha-2 country code of this business.
26	location.cross_streets	string	Cross streets for this business.
27	location.display_address	string[]	Array of strings that if organized vertically ...
28	location.state	string	ISO 3166-2 (with a few exceptions) state code ...
29	location.zip_code	string	Zip code of this business.
30	messaging	object	Contains Business Messaging / Request a Quote ...
31	messaging.url	string	Action Link URL that drops user directly in to...
32	messaging.use_case_text	string	Indicates what kind of messaging can be done w...
33	name	string	Name of this business.
34	phone	string	Phone number of the business.
35	photos	object[]	URLs of up to three photos of the business.
36	price	string	Price level of the business. Value is one of $...
37	rating	decimal	Rating for this business (value ranges from 1,...
38	review_count	int	Number of reviews for this business.
39	url	string	URL for business page on Yelp.
40	transactions	string[]	A list of Yelp transactions that the business ...
41	special_hours	object[]	Out of the ordinary hours for the business tha...
42	special_hours[x].date	string	An ISO8601 date string representing the date f...
43	special_hours[x].is_closed	boolean	Whether this particular special hour represent...
44	special_hours[x].start	string	Start of the opening hours in a day, in 24-hou...
45	special_hours[x].end	string	End of the opening hours in a day, in 24-hour ...
46	special_hours[x].is_overnight	boolean	Whether the special hours time range spans acr...
47	attributes	object	A mapping of attribute names, such as "Ambienc...