**Predicting Target Variable from Feature Variables**

Our model is going to predict star rating for businesses in North America based on the features such as location, business categories, etc.

# **Introduction**

Yelp is a website which publishes crowd sourced reviews about local businesses (Restaurants, Department Stores, Bars, Home-Local Services, Cafes, Automotive). It provides opportunity to business owners to improve their services and users to choose best business amongst available. Our objective here is to create the best possible model to accurately predict the business star rating. This prediction model can help business owners to understand the market better. With a dataset of different business categories and their Yelp ratings, we decide to use a Multiple Linear Regression model to investigate what factors most affect a business’s Yelp rating and predict the Yelp rating for an owner’s business.

# **Data Acquisition and Organization**

This research will be based on the statistical data provided by Yelp (Yelp Open Dataset). The dataset includes information about 160,585 local businesses in 10 metropolitan areas. Currently, the metropolitan areas centered on Montreal, Calgary, Toronto, Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland, are included in the dataset. It comprises of 6 JSON files-

1) business.json, which comprises of business-related data including location, attributes, categories, average ratings, hours open etc.

2) review.json, which contains full review text and rating for a business by a user, stars, votes - which indicate how helpful a review was, etc.

3) user.json, which includes users and their friends mapping, total reviews and average stars by that user, votes received by other users etc.

4) checkin.json, which records hourly check-ins for businesses

5) tip.json, which contains short tips written by users for that business which would convey quick suggestion

6) photos.json, which stores photos posted for businesses.

We will do our modeling on the business.jason dataset.

# **Exploratory Data Analysis**

***Data Classification***:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Description** | **Data Type** | **Feature Type** |
| business\_id | 22 character unique strin | string | Categorical |
| name | the business's name | string | Natural Language |
| address | the full address of the business | string | Location data/Natural Language |
| city | the city | string | Categorical |
| state | 2 character state code, if applicable | string | Categorical |
| postal\_code | the postal code | string | Categorical |
| latitude | latitude | float | Numerical |
| longitude | longitude | float | Numerical |
| stars | star rating, rounded to half-stars | float | Numerical |
| review\_count | number of reviews | integer | Numerical |
| is\_open | 0 or 1 for closed or open, respectively | integer | Categorical |
| attributes | business attributes to values. note: some attribute values might be objects | object | Categorical |
| categories | an array of strings of business categories | object | Categorical |
| hours | an object of key day to value hours, hours are using a 24hr clock | object | Date-time |

***Dimension of the dataset:***

Table

Description automatically generated

***Descriptive Statistics:***

Descriptive statistics can give one great insight into the shape of each attribute.

Table

Description automatically generated

***Here are some plots showing the relationship between variables:***

**Distribution of stars:**

Chart, histogram

Description automatically generated

**Scatterplot between Latitude and Longitude:**

Chart, scatter chart

Description automatically generated

**Yelp star Histogram:**

Chart, bar chart

Description automatically generated

## Data Cleaning and Feature Engineering:

At the beginning, we took one independent feature ‘latitude’ to train the model, while “stars” was our target variable. Next, we included ‘longitude’ with latitude and observed how the Regression coefficients get affected. We extracted ‘state’ and ‘city’ features one at a time and then performed get\_dummies as our method of encoding categorical features. The last feature we worked on was “categories”; there were so many different types of business categories, we performed Count Vectorizing on that feature.

**Checking for null values:**

Table

Description automatically generated

We checked for null values, there were 115 for the “categories” feature. Since these are business categories, we cannot impute the missing values, thus we will most certainly need to drop these observations.

Graphical user interface, text, application, email

Description automatically generated

**Encoding of “state” and “city” columns:**

**Table

Description automatically generated**

**There’s a total of 31 states and 836 cities, so total 867 columns will be added to the business dataset because of the encoding.**

**Getting rid of leading and trailing spaces and lowercasing the categories:**

**We take a look at the first element of the ‘categories’ feature, we see that it is a string. But for us to be able to apply a vectorizer, we need to turn this into a list. This is how it looks before conversion into a string:**

**Graphical user interface, application

Description automatically generated We are defining functions to turn the different business categories into lowercase strings and t get rid of extra spaces.**

**Text

Description automatically generated with low confidence**

**Graphical user interface, text, application, email

Description automatically generated**

**Text

Description automatically generated**

**Count Vectorizing:**

**Table

Description automatically generated with medium confidence**

**Modeling:**

Several models have been used with various feature selection. The model was getting more accurate while using more of the available features.

**Below is the table of all the models we trained. We are comparing the test set accuracy and the training set accuracy for each model so we can see what cause the model to improve. We tried to answer the following questions:**

**1. How well the model did**

**2. Was it good producing an accurate model?**

**3. Was it better when we incorporated more features? Or worse?**

**4. Was it better on our training set and our test set, or one or the other?**

**Here’s how the hyperparameter table looks like:**

**Hyperparameter table with Sorted values:**

**Graphical user interface, text, application

Description automatically generated**

**Hyperparameter table with description:**

**Table

Description automatically generated**

***Model Summary:***

**I learnt that latitude has a very week positive correlation with the target “stars” variable (0.00003). As soon as I normalize for the “longitude”, they both have negative correlation with reviews/stars value. As we normalize for more variables, those coefficients change.**

Graphical user interface, text, application

Description automatically generatedThe coefficient is positive when we included only “latitude” as a feature. , which indicates if we move further North, businesses in general tend to receive very slightly more positive reviews (not that significant though).

Graphical user interface, text

Description automatically generatedThe coefficients were negative when we included “longitude” in the second model. This means if we move further North, businesses in general tend to receive -0.005 star-rating/deg. For example, businesses in NE, tend to do slightly worse than those is TX. Similar, moving further east, causes a decline by 0.004 star-rating/degree.

When we extracted features “latitude”, “longitude” and “state”, Table

Description automatically generatedfrom the coefficient values we see that most positive slopes are for IL, TX, FL, MN, etc. These states are likely to have the highest star ratings for all businesses. The worst case finding seems to be for Wyoming since it has the most negative value. For WY, VA, BC, businesses get an average value of 2 stars less than the average for all states. And they get almost 4.5 stars less than for TX, IL, etc.

State is an exclusive feature. One can be in one state at a time. We need to choose one state at our end and will observe the star values that we lose or gain in our state (average) or another state. We will choose TX as the reference. Reason for Wyoming to have the lowest star ratings in comparison to other states may be there aren’t many businesses there. Not too many urban developments. It’s also possible that Yelp may be running a flawed algorithm to analyze content, that wrongfully filters out good reviews as fraudulent ones- on any business from specific states, like WY. Or, probably not too many businesses in those states use social media overall.

It’s also possible that our model may be ignoring some parameters. Or may be the few business that are in those states are just bad business. The ones that have lots of good rating, are pretty urban. Like Chicago is the biggest city in Illinois, a lot of business show up in Chicago, likewise in TX. Also, Florida has a lot of vacation spots, people are generally happy when they visit a restaurant or a business facility, so for high tourism states, star ratings are given more generously by consumers. Hence more positive coefficients compared to other states.

Form the hyperparameter table we see that the best accuracy (least root mean square for test data) is yielded when we work on features ‘latitude’, ‘longitude’, ‘state’.

**The top and bottom five model coefficients look like this when we used the features** ‘latitude’, ‘longitude’, ‘state’, and ‘city’.

**Table

Description automatically generated**

**Future Work:**

We can do sentiment analysis using natural language processing and machine learning techniques with the help of Yelp’s business.json and review.json datasets. This is an important problem to solve as it will save time of an individual while making decision about the product to solely allow for checking the ratings instead of textual reviews. From the business.json data, we can identify some interesting features like **name** and **address** and consider them as natural language data. Address is a location data, but can also be treated as natural language. The **name** feature, although a string, is pretty unique across different businesses. Individual words will not be unique. Many restaurants will have names that contain same words, we can extract those words and find those words that are highly correlated with high reviews. We can use vectorizers on Textual Features, like Count Vectorizing, TFIDF (Term Frequency Inverse Document Frequency)) vectorizing, or any other text to vector embedding as part of feature engineering on Natural language data.

**Using the**  Yelp Dataset, we can create a model that evaluates whether a restaurant is likely to succeed or fail within the next couple of years. It can help restaurant lenders (such as banks) and investors decide whether they should lend/invest at a particular restaurant based on the likelihood that it is going to fail within that specific time frame. We can find a list of restaurants that existed at some point in the past and then match that information with current information about the restaurants. We can extract the “**name**” feature and If the restaurant name appears more than once in the list then we can considered to be part of a chain and suggest that chains are likely to be closed. Based on the restaurant coordinates, we can created a list of restaurants within 1 mile radius for each of the restaurants in the list.

* We can extract **review\_count**, **star** features to compare the star rating and count of reviews relative to surrounding restaurants.
* Also, we can determine the **age** of a restaurant by the date of the first yelp review. For that we need to join business.jason dataset with review.jason dataset on the business\_id key.

We can examine fake review detection data, which consists of 350,000 user reviews. The true and fake reviews in the data set help us train a model that predicts if a given review is fake or not.

**Conclusion:**

In conclusion, we have experimented with various feature selection and supervised learning algorithms to predict star ratings of the Yelp dataset using business.json file of the Yelp Dataset. We evaluate the effectiveness of the machine learning algorithm based on accuracy by Root Mean Square Error. We pick a feature, extract it, clean it up, make it numerical, added to our model. We do testing and validation on the model and then interpret the results. The hyperparameter tuning that we did here was Feature Engineering. First, we added ‘latitude’ as our independent variable, did a Linear Regression on that, got the coefficients, we did testing and validation on that and kept on adding more interesting features.