**Restaurant Entrepreneur: Predicting best type and location with Yelp Review Data**

Bill (an investor) went on a work trip recently to Las Vegas and Phoenix for a few days. During his stay, he really liked the restaurant options that were available. Being that he already had interests in opening up a restaurant for some time now, he wanted to know what the type of restaurant and where to open would be most successful.

During Bill’s work trip, he used Yelp to search for different restaurants near the hotels he stayed at. Bill was thinking about opening a restaurant in either Las Vegas or Phoenix, because he was impressed with the variety of great food options available. Before he opens a restaurant, he is interested in learning more about what types of restaurants are in these cities, what are the yelp stars, and additional details that would help him make a decision where to open restaurant.

As a first look, our group decided to use Yelp API to collect data on the mix of restaurants across cities and states within the United States. In this case, we decided to use the review data that Yelp is known for. Since Yelp provides a wide variety of restaurant categories, it is possible to get more information related to reviews and ratings for all different types in each city.

After our initial attempts to use Yelp API, we found issues with importing the Yelp API data. When we created a dataframe from the API, it would only pull a small portion of the data on each run, not a sufficient amount of data to do a full analysis on. Fortunately, we were able to find some recent Yelp Review data on Kaggle.com that had a huge dataset to work with. The dataset included detailed information of the business such as, restaurant ID, location, postal codes, star ratings, review count, etc.

**Why we chose these specific models and how do they work specifically with our dataset?**

* Before deciding on which Machine Learning Model to use to conduct our analysis on, we wanted to test a few to see which one would be the best. We chose Deep Learning, Random Forest Classifier and Logistic Regression as our Machine Learning Models. Random Forest Classifier is a good model if you want high performance with less need for interpretation. Deep Learning Model is known for its supremacy in terms of accuracy when trained with huge amounts of data and to get more neural network predictions. Logistic Regression is most useful when we want to predict the probability for a categorical response variable with two outcomes. In our case, we’re trying to decide on what category type of restaurant and where is best to open based off of star ratings. A good rating would be any scores between 3-5, bad would be 0-2.
* After our initial attempts with a few machine learning models and getting very low accuracy scores on each with multiple attempts, we needed to test our data on other alternative models. We tried Linear Regression and Random Forest Regressor as suggested by our professor and teacher’s aid.

**Detailed description of preliminary data preprocessing**

* We downloaded our data from Kaggle.com which was a json file and cleaned it. The cleaning process was filtering on only the data we needed, so we dropped a bunch of data, such as restaurants that were already closed or rows with null values. We filtered out all restaurant businesses based on category column into a new dataframe. Because we wanted to know what the best category type of restaurant to open was, we explored to see how many unique categories of restaurants were in the dataset, then created a new column called “ethnic\_type” to put all the categories needed for our analysis. This new column is a feature we needed to add, to separate all the options we had available to decide on. We chose 20 unique restaurant types which were African, American, Asian\_Fusion, British, Chinese, French, Greek, Hawaiian, Indian, Italian, Japanese, Korean, Mediterranean, Mexican, Middle\_Eastern, Spanish, Thai, and Vietnamese.
* We did a value count for each city to see if there was enough data for Arizona and Nevada, which showed to have more than 1500 businesses for Las Vegas and Phoenix. This confirmed that we had enough data to conduct our analysis.
* With this cleaned data, we created a new dataframe and then uploaded to postgres with a connection through SQL. In addition, we created CSV files of the cleaned data.
* We imported StandardScaler and get\_dummies. The StandardScaler is needed to transform the data so that it has a mean of 0 and a standard deviation of 1. The get\_dummies is needed as it creates a binary column for each category type of restaurant.
* We've also imported train\_test\_split which will help us split our data for training and testing.
* Further in the analysis process, we then added another column, which showed a prediction of star ratings, which were the review ratings on the graded scale of 0-5 stars on how satisfied customers were for each restaurant/business.

**Description of preliminary feature engineering and preliminary feature selection, including their decision making process**

* Initially, we started with Linear Regression Model as one of the 3 models we wanted to test and soon realized that we can only use this model when there are continuous values to be predicted. For categorical data prediction, Logistics Regression Model should be used. After poor results from Logistics Regression Model, our professor suggested to go back and use Linear Regression Model. This still ran into issues with this.
* Our goal is to predict the star rating, hence our attempts to find the models that could fit our output as well. While analyzing each Machine Learning model that we have tested, our Random Forest and Deep Learning samples seemed to be closer to our requirements.

**How was the data split into training and testing sets?**

* The training and testing datasets were divided with 67% going towards training and 33% going towards testing.

**What was our model’s accuracy? What were the limitations and benefits of each model we chose?**

**Deep Learning**

Benefits – An advantage is its capacity to execute feature engineering on its own. A deep learning algorithm will scan the data to search for features that correlate and combine them to enable faster learning without being explicitly told to do so. Another advantage is they produce the best results with unstructured data. Most company’s data is unstructured because of the different formats they all come in from. Unstructured data is hard to analyze for most machine learning models. Deep learning algorithms can be trained using different data formats, and still deliver good insight that is relevant to the purpose of its training.

Limitations – It needs a large dataset to go through to predict the best outcomes, just like the human brain needs a lot of experiences to learn and deduce information before making any decisions. Overfitting is also another negative for the Deep Learning Model as it can train the data too well. Overtraining is a problem in neural networks. You can tell when a model is overtrained when the accuracy % stops improving after a certain number of epochs and flattens out.

For the dataset that we are using with ethnic type and city as our categorical data, our accuracy stayed around 40%. We had thought maybe our dataset was too small, so we ran a test with fabricated data 3 times larger than our original data to see if that would make a difference, and the accuracy stayed around 40%. This machine learning model was not a success.

**Random Forest Classifier**

Benefits – There is very little pre-processing that needs to be done. The data usually does not need to be rescaled or transformed. Predictions and training speeds are much quicker.

Limitations – For large datasets, they take up a lot of memory. They also tend to overfit.

For the dataset that we are using with ethnic type and city as our categorical data, our accuracy was at best 42%. The challenge that we think is that we are trying to train the entire dataset just with category and number of reviews per rating. This makes the dataset limited per category for it to be trained.

**Logistic Regression**

Benefits – It is easier to implement, interpret and very efficient to train. It gives an easy measure of how relevant a predictor is and it’s direction of association (positive or negative).

Limitations – It cannot solve non-linear problems.  It heavily relies on a proper presentation of your data. This means that logistic regression is not a useful tool unless you have already identified all the important independent variables. Since its outcome is discrete, Logistic Regression can only predict a categorical outcome. It is also an Algorithm that is known for its vulnerability to overfitting.

The accuracy rate in our analysis using this model is still 30%, which shows that the data is underfitting.

**Linear Regression**

Benefits – Implementation is simple as it’s algorithm is the least complex compared to others.

Limitations – Lack of practicality and most problems in the real world aren’t “linear”. It assumes that there is a straight-line relationship between the dependent and independent variables which is incorrect most of the time.

We attempted Linear Regression with X = zip-code and Y = review count or stars. We then grouped the data by zip-code. Our intention was to get the star rating by zip-code per prior review, but we were not able to achieve this.

**Random Forest Regressor**

After multiple attempts with all the other Machine Learning Models, and only achieving low accuracy scores, we tried Random Forest Regressor as per suggestion by our professor and teacher’s aid. This method gave a significantly better result at 90% accuracy. With every other machine learning model’s accuracy score being less than half of what we were able to achieve than with random forest regressor, it was a no brainer to move forward with this machine learning model.

Regression algorithms attempt to estimate the mapping function from the input variables (x) to numerical or continuous output variables (y). In this case, y is star rating and hence a real value. This could be an integer or a floating point value. Initially we tried with classification model considering there are multiple input values and the expected output as 1 through 5. We considered these as classifiers, but by doing so, the classifier was probably not distributing the weight correctly. This was why the accuracy could not get beyond 45%. Upon considering changing the model to include the output as continuous variable, the accuracy score seems to have improved to around 90%.

To further improve on this model, we suggest the below:

1. Remove zip-code from the modeling
2. Search for more data
3. Use text data as one of the features
4. Change the prediction as good and bad instead of percentage using text data