**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.

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 in regards 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.

Before deciding on which Machine Learning Model we were going to choose to do our analysis on, we wanted to test a few to see what would be the best. We began setting up our code by importing various libraries, which included Random Forest Classifier and DeepLearning 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 it's supremacy in terms of accuracy when trained with huge amounts of data and to get more neural network predictions. We've also imported train\_test\_split which will help us split our data for training and testing.

For the preprocessing, we imported StandardScaler and OneHotEncoder. The StandardScaler is needed to transform the data so that it has a mean of 0 and a standard deviation of 1. The OneHotEncoder is needed as it creates a binary column for each category type of restaurant.

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

* We chose Deep Learning, Random Forest Classifier and Logistic Regression as our Machine Learning Models. Random Forest is a good model for high performance with less need for interpretation. Deep Learning is known for it’s 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 reviews. A good review would be any scores between 3-5, bad would be 0-2.

**Detailed description of preliminary data preprocessing including preliminary feature engineering and feature selection**

* 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.
* 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.

**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 – One main advantage is it’s capacity to execute feature engineering on it’s 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’s relevant to the purpose of it’s 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.

**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.

**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.