Capstone Two

MACHINE LEARNING FORECAST MODEL

IDENTIFYING LUXURY VEHICLE FLEET DEMAND

MARTELL TARDY

06/30/2021

Report - IDENTIFYING LUXURY VEHICLE FLEET DEMAND

1. Introduction

1.1 Problem

Lexus of Mishawaka is an authorized Lexus dealership conveniently located on Grape Road in Mishawaka, Indiana. Lexus of Mishawaka is a full-service dealership offering their guests not only a full line-up of all new and L/Certified Lexus vehicles, numerous luxury and mid-range vehicles from similar brands, but also a friendly and reliable service and parts department. In order to ensure Lexus of Mishawaka is accurately serving their market and the needs of their guests', Lexus of Mishawaka has tasked their in-house Marketing and Information Specialist, Martell Tardy, with the task of analyzing their 2004 -2017 historical data for insight.

1.2 Criteria for Success

Success for this project would be the training and deployment of a machine learning model that will be able to forecast which Lexus, Toyota, and non-Toyota models are necessary to have in the dealership inventory 24 months starting April 2017. This forecast will improve dealer order and inventory management, optimize plant production scheduling, and increase understanding of consumer demand in the market.

1.3 Dataset

The dataset for this project was collected directly from the Lexus of Mishawaka Principle, Perry Watson III via their CRM system, VINSolutions in March 2017. The data consists of 8,208 entries of actual sales history which contains the following but not limited to information:

- Contract Date
- Vehicle Make
- Vehicle Model
- Contract Term
- Vehicle Sale Price
- Trade Vehicle Information

2. Data Wrangling

Prior to data wrangling, I utilized json_to_csv.py to flatten nested json objects within a business' attributes column which has created additional 40 columns that have 'attributes' as its prefix. For example, attributes.Caters, attributes.RestaurantsAttire, etc.

The scope of this project is geared towards 'independent' restaurants in the hospitality industry, therefore, businesses' dataset needed to be cleaned and transformed. Below are the following data cleaning summaries per dataset.

2.1 Business Dataset

The business dataframe contained 209393 businesses mainly from hospitality industry (restaurants, bars, food, etc.), however there were significant number of businesses that were part of chain restaurants/businesses and businesses that were not food/drinks related (law firms, pet grooming, real estate, etc.); therefore, those were needed to be filtered out.

- Removed 52954 businesses that were part of chain restaurants which cut reduced to 143,958 (38% reduction)
- Removed non-food/drinks related businesses reducing from 143,958 to 44,046 businesses (69.4% reduction)

Handling NaNs in Business Dataframe

There were multiple columns related to businesses' attributes such as goodforkids, goodforgroups, tableservice, delivery, etc. that had NaNs - as it was not explicitly defined on the initial dataset - I had set it to 0 as if it did not offer those services.

Missing price value within the price column (16%) is replaced by rounded average price from most similar businesses between (1-4).

Summary of Business Dataset

1. Filtered business dataset to only populate hospitality related businesses (restaurants, clubs, bars, etc.).
2. Removed businesses that were missing both attributes and categories values as those will be essential for feature engineering.
3. Removed multi-unit restaurants such as chain restaurants as this project is concerned with single-unit businesses.
4. Used categories and attributes data to identify restaurants' characteristics and what type of food/cuisine they are serving.
5. Used cosine similarity amongst businesses to fill in missing price range.
6. Removed columns that were not needed such as address, city, state, and etc.
7. Increase column size from 60 to 144 columns

8. Adjusted average star rating to have better representation at the same scale. For instance, one restaurant with 5 stars rating with 2 reviews is not the same as another restaurant with 3.5 stars with 100+ reviews.
9. Reduced business dataset from 209,393 to 44,046 rows.

2.2 Review Dataset

The review dataframe contains 8,021,122 entries.

- Removed 4,750,990 reviews that were part of businesses in filtered business dataframe.
- Convert text column from object to string datatype

Summary of Review Dataset

1. Reduced business dataset from 8,021,121 to 3,270,132 rows (59% reduction); using filtered business dataframe's business_id column.
2. No NaNs found
3. Added two new columns:
   - review_type: defines whether review is positive or negative in binary value based on the user's average star rating.
   - text_count: total text count per review

2.3 User and Tip Dataset

The user dataframe contains 1,968,703 entries which reviewed businesses spanning across 10 metropolitan areas. For this classification project, a user dataframe is not needed as the column values provided do not bring any value in predicting whether a restaurant will close or not. User dataframe is used to filter tips dataframe using filtered user dataframe (removed users that were not in filtered review dataframe.)

Unlike review dataframe, tips does not have any quantifiable values to determine whether it is a good comment or not for the restaurants. Therefore, sentiment analysis was used to define each tip as positive (1) or negative (0) based on its compound score.

Note

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

- positive sentiment : (compound score >= 0.05)
- neutral sentiment : (compound score > -0.05) and (compound score < 0.05)
- negative sentiment : (compound score <= -0.05)

Summary of Tip Dataset

1. Converted tip's text column datatype to string.
2. Filtered tip dataset using updated user dataset's user_id.

3. Removed all columns except user_id, text, date, and sentiment analysis related columns.
4. Reduced tip dataset from 1,320,761 to 1,136,880 rows. (14% reduction)
5. Removed two rows that had null values.

2.4 Check-In Dataset

The check-in dataframe contains 175,187 entries which is the least amount of entries compared to aforementioned datasets.

Summary of Check-In Dataset

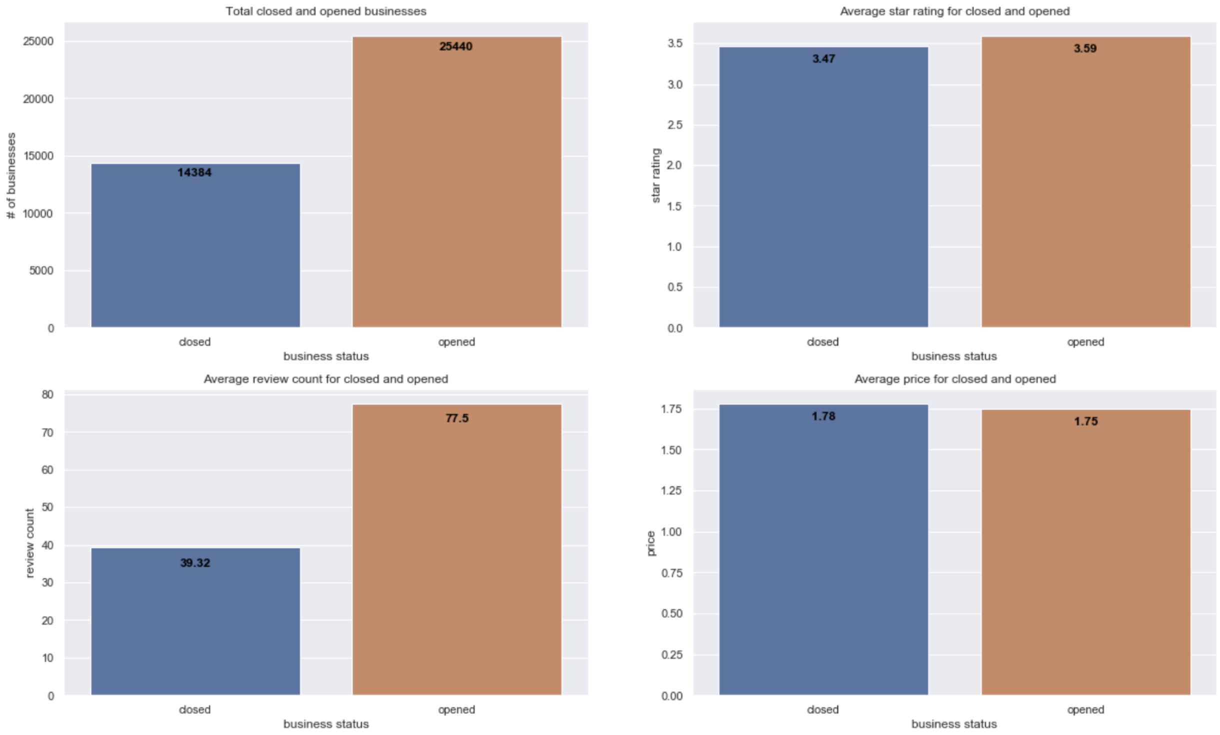
1. Filtered check-in dataset using filtered review dataframe
2. Reduced check-in dataset from 175,187 to 42,296 rows.
3. No null values were found

3. EDA

3.1 Visualize closed and open restaurants' attributes

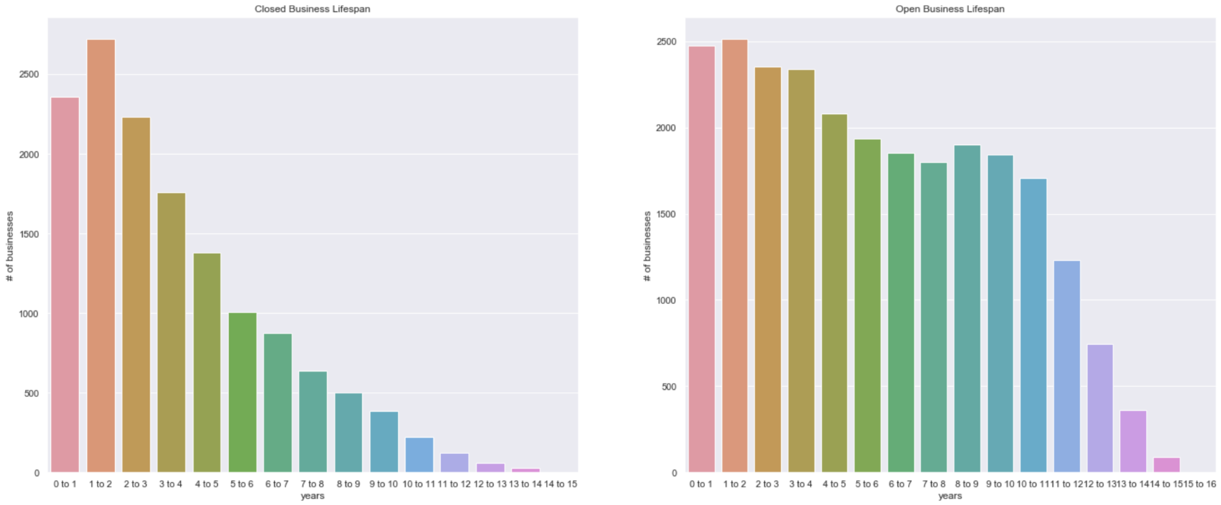
- Average star rating,
- Average price
- Average review count
- Total count of closed and open restaurants

Open businesses tend to have higher average values across the board. One interesting finding from these bar graphs are that the average review count is significantly higher for open businesses than closed businesses which makes me hypothesize that more reviews means more foot traffic to the restaurant (higher chance of remaining open), which using this feature may yield considerable weight in accurately predicting restaurant is likely to be successful during machine learning part of the project.



3.2 Closed and open business count by lifespan

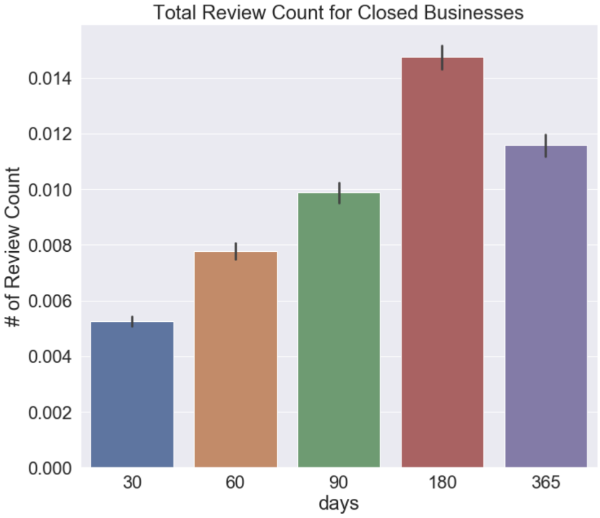
Majority of the businesses fail within the first 3 years; closed businesses' downward trend is more apparent compared to open businesses. Open businesses tend to have a lot more businesses spread out in all lifespan categories.

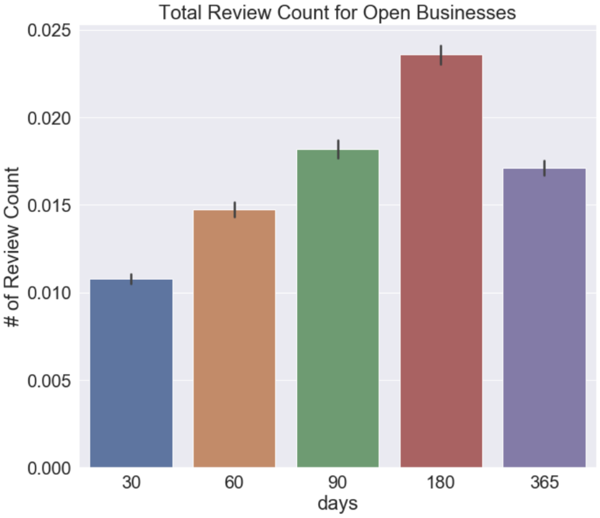


3.3 Review count trend in last 30, 60, 90, 180, and 365 days

- Scaled column values for 30 ,60 ,90 ,180 , and 365 days

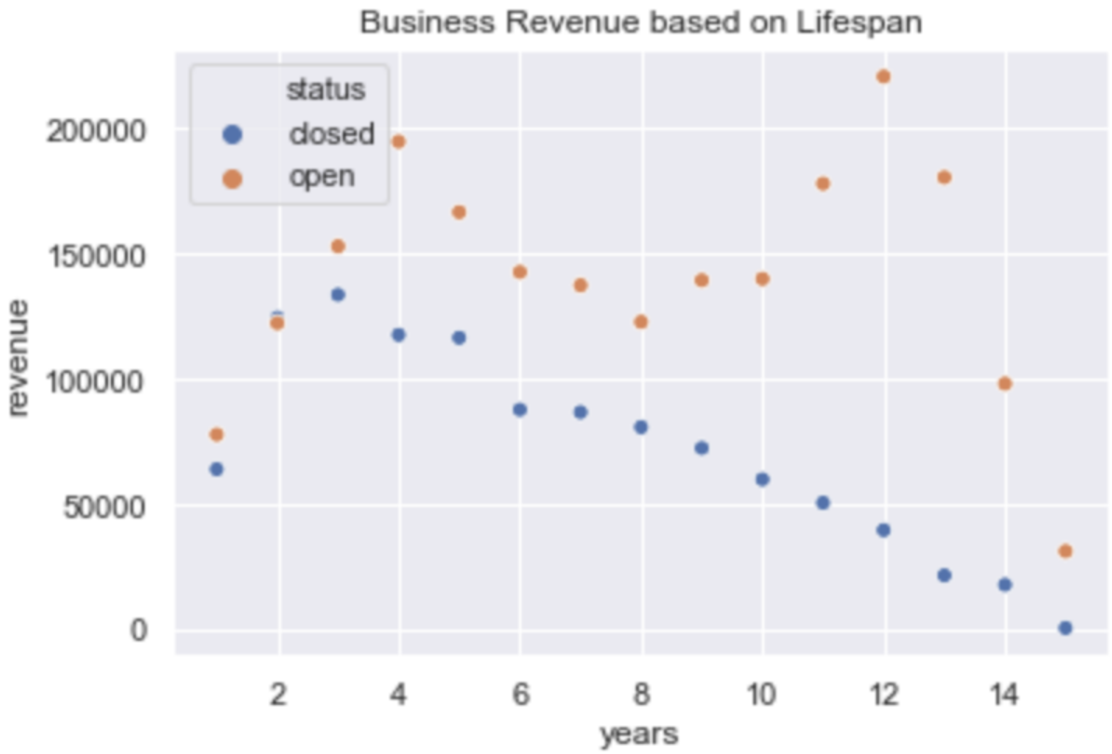
Both closed and open restaurants had an upward trend in getting more review counts over the last one year. The only difference between two is that open restaurants generally have more reviews.





3.4 Businesses' revenues vs age

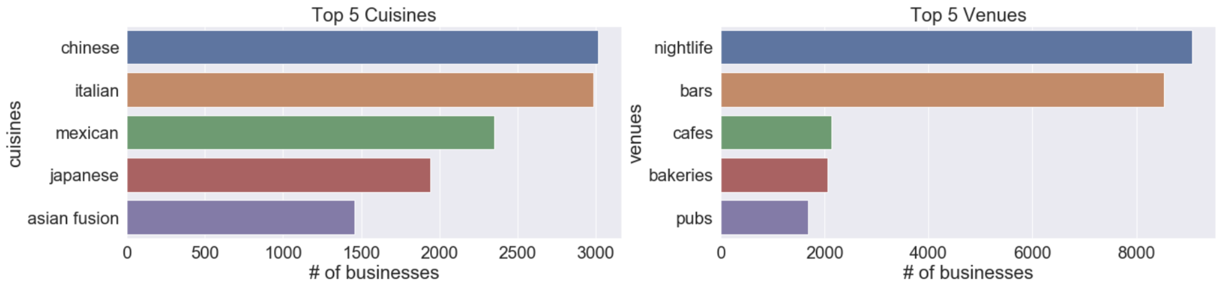
After random sampling from open businesses, currently opened restaurants did consistently better than its closed businesses in terms of revenue. Closed restaurants had a downward trend in sales as the year went on.



3.5 Top 5 Cuisines and Venues

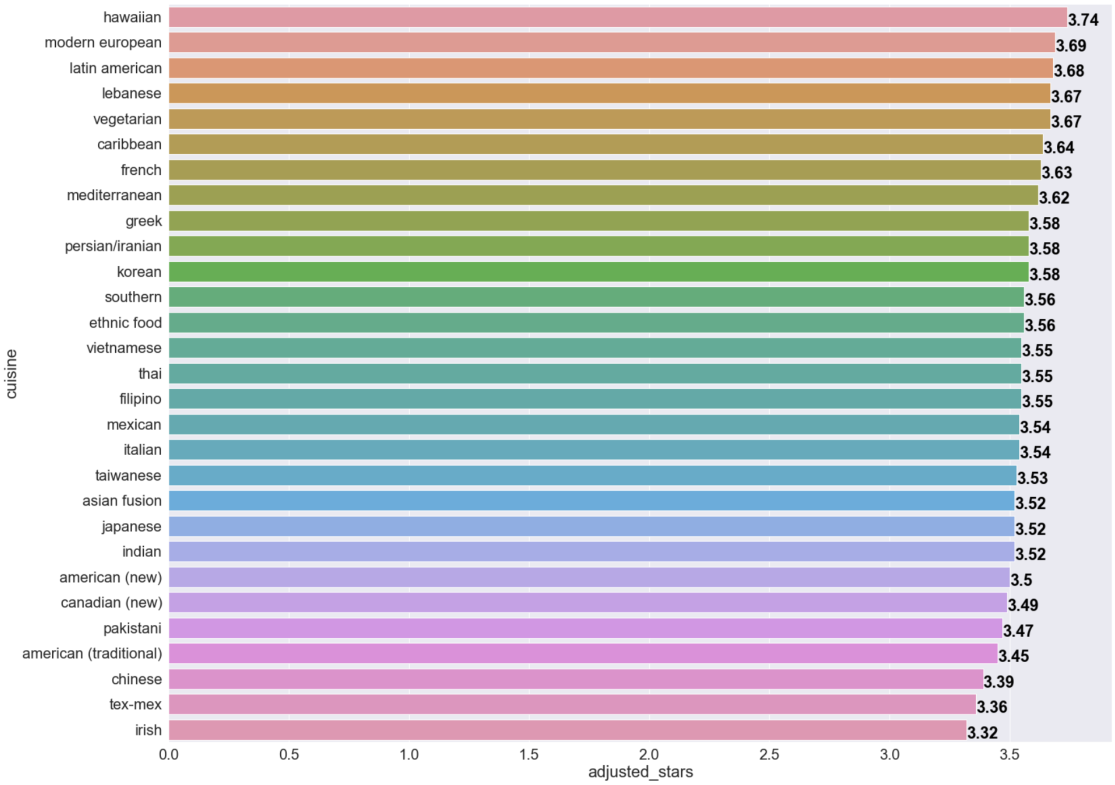
I intentionally omitted 'American food' from cuisine and 'Restaurants' from venues as they were overwhelmingly common throughout the business dataframe. I wanted to visualize other top cuisines and venues aside from aforementioned cuisine and venue.

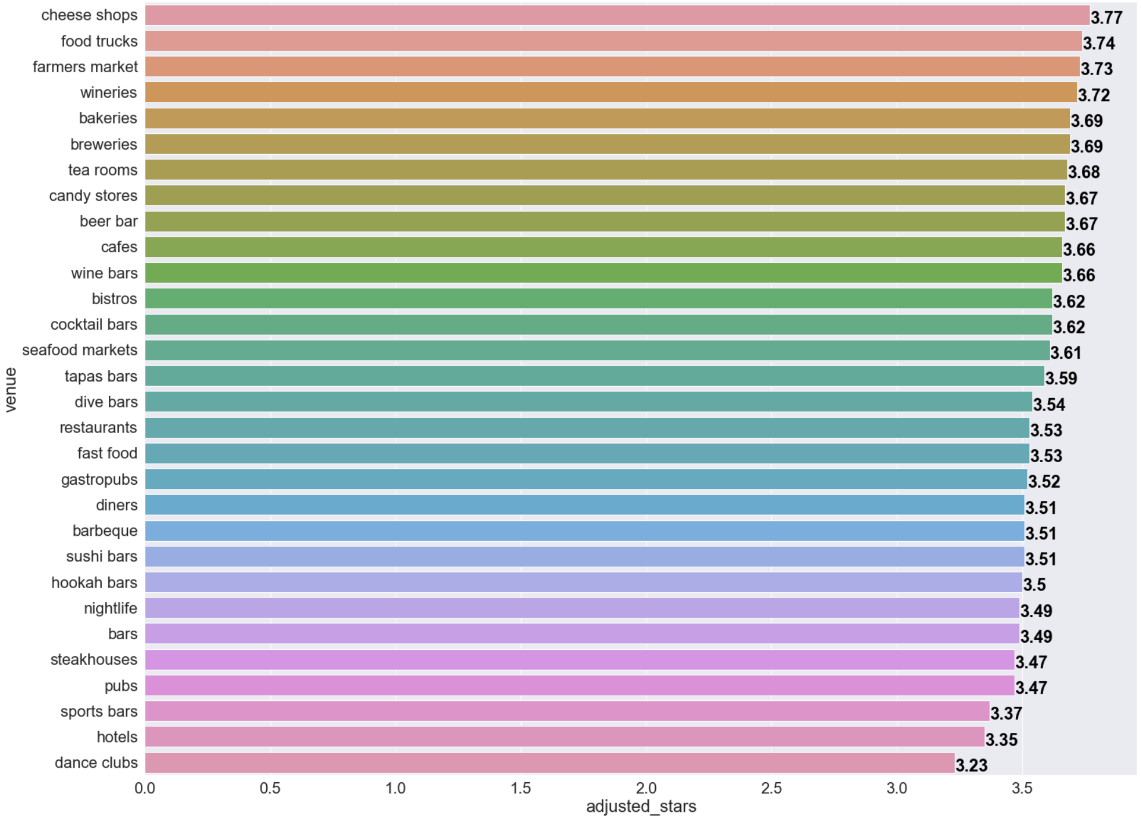
Mexican, Italian, Chinese, Asian fusion and Japanese food are the most common cuisines and common venues are nightlife, bars, caes, bakeries, and pubs.



3.6 Cuisine and Venue Ranking by Average Star Rating

Visualizing top cuisines and venues based on star rating.





Top 5 venues based on star rating are cheese shops, food trucks, farmers market, wineries, and bakeries; while top 5 cuisines based on star rating are hawaiian, modern european, latin american, lebanese, and vegetarian.

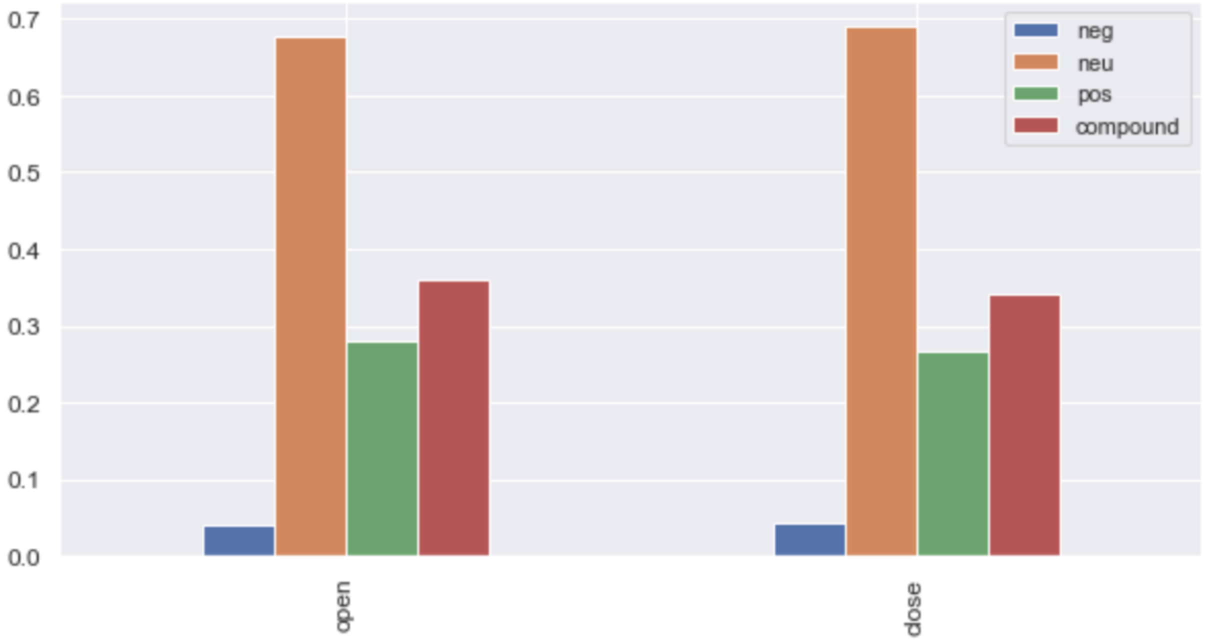
One of the common correlations between those two results above are that top 5 mentioned venues and cuisines do not have significant amounts of review counts compared to the bottom half of the list. Therefore, generally, more reviews meant lower overall star rating.

3.7 Reviews' and Tips' Sentiment Analysis

Visualizing sentiment analysis score for both closed and open restaurants.

The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

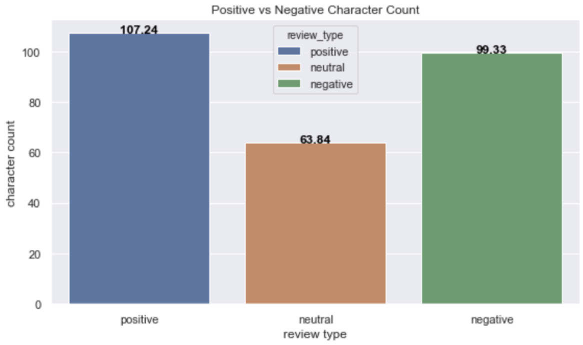
- positive sentiment : (compound score >= 0.05)
- neutral sentiment : (compound score > -0.05) and (compound score < 0.05)
- negative sentiment : (compound score <= -0.05)

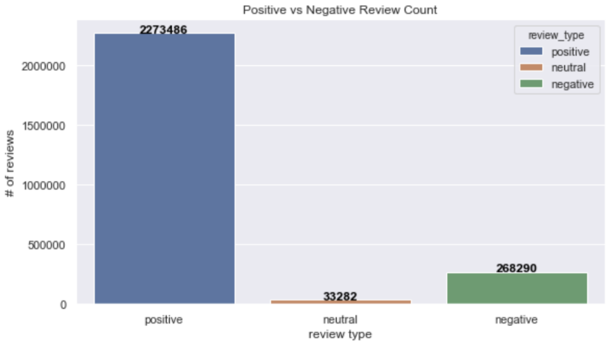


Generally, open restaurants have higher compound and positive sentiment scores while close restaurants have higher negative and neutral sentiment scores. Therefore, how users/customers perceive the restaurants have some impact on whether restaurants will stay open or not.

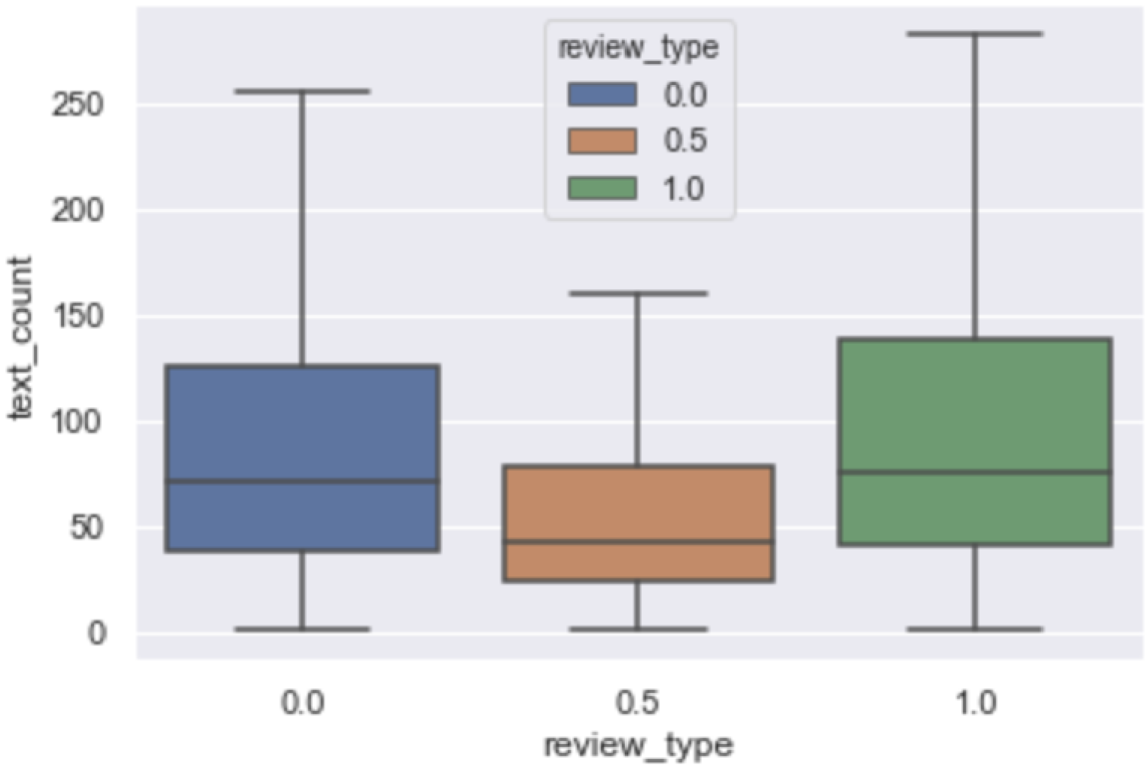
3.8 Total positive, neutral, negative review count and its average text count

Check how many reviews are positive, neutral, and negative. Identify average text count included in positive, neutral, and negative review.



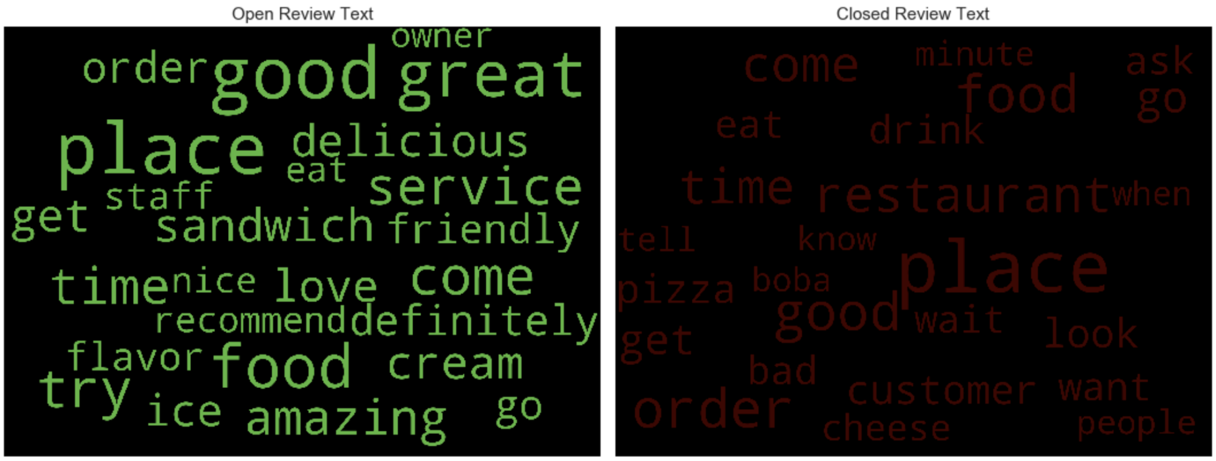


Identifying text count through box plot to see count from 25%, 50%, and 75% quartiles.



Majority reviews are positive and had slightly higher average text count compared to neutral and negative reviews. It is interesting that bulk of reviews are positive; this makes me think that customers are more willing to leave reviews when they're satisfied with restaurants' service and/or food, inversely when customers are not satisfied with it, majority of the customers does not leave any feedback for restaurants to improve their service and food.

3.8 Visualize word cloud for top 10 highest and lowest ranked businesses


Open Review Text

Closed Review Text

Considering the majority of the reviews are positive, there doesn't seem to be much difference between open and closed reviews. However, open review wordcloud tends to have more obvious positive adjectives compared to closed businesses' reviews.

4. Prepare and create model data to be used for statistical analysis and machine learning

- Updated Business dataframe by adding sentiment analysis score, user_review_type and updating revenue accumulated from tips and reviews dataframe.
- Got total star rating count (star_1.0, star_2.0, etc.) per restaurant
- Added two new columns pos_reviews and neg_reviews - which shows total positive and negative review counts for each restaurant based on comparing the review's star rating and user's average rating. When a star rating is higher than a user's star rating, it is considered a good review and vice versa.
- Used checkins' total count to update each restaurant's revenue by multiplying check-ins with the restaurant's price range.
- Define positive and negative reviews based on user average rating
  - Every user rates differently and star ratings can be subjective; therefore, to minimize subjectivity, I am defining whether the review is positive or negative relative to the user's average rating. For example, if a user rated one restaurant with 2.5 but has the given average rating of 2.1 stars, I am defining it as positive since it's above the average.
- Got total star rating count (star_1.0, star_2.0, etc.) per restaurant
- Removed name, business_id, stars, and state columns from business dataframe
- Utilized MinMaxScaler to normalize value between 0-1 to all columns to improve convergence speed during machine learning.

Cleaned (test) dataset

The cleaned dataset is used for statistical analysis and machine learning. It contains 39,824 rows with 158 columns. All columns have float64 data-type ranging from 0 to 1.

'test' dataframe

The test dataframe is sparsely defined since the majority of columns are restaurants' attributes and cuisines such as 'tableservice', 'drivethru', 'french', 'korean', 'casual', and etc. It also contains other added columns such as restaurants' lifespan, sentiment analysis scores, star rating count, adjusted_star (count) etc. to help predict whether a restaurant is prone to failure or not.

Features

Below are the following main features (not restaurant attributes):

1. **review_count**: defines total reviews given to the restaurant
2. **adjusted_stars**: adjusted star rating with the merit of incorporating both average restaurant rating (positive) and number of ratings (high traffic).
3. **price**: Restaurant price range.
4. **30_days_review_count**: Reviews received by the restaurant in the last 30 days.
5. **60_days_review_count**: Reviews received by the restaurant in the last 60 days.
6. **90_days_review_count**: Reviews received by the restaurant in the last 90 days.
7. **180_days_review_count**: Reviews received by the restaurant in the last 180 days.
8. **365_days_review_count**: Reviews received by the restaurant in the last 365 days.
9. **lifespan**: Restaurant's business life span (how it has been operating for).
10. **neg**: Restaurant's negative sentiment score.
11. **neu**: Restaurant's neutral sentiment score.
12. **pos**: Restaurant's positive sentiment score
13. **compound**: Restaurant's compound sentiment score (overall score)
14. **revenue**: Restaurant's revenue (calculated through tips' and reviews' count multiplied by restaurant's price).
15. **pos_review**: Restaurant's positive review count (based on user's average score)
16. **neg_review**: Restaurant's negative review count (based on user's average score)
17. **stars_1.0**: Total number of times the restaurant received 1-star rating.
18. **stars_2.0**: Total number of times the restaurant received a 2-star rating.
19. **stars_3.0**: Total number of times the restaurant received a 3-star rating.
20. **stars_4.0**: Total number of times the restaurant received a 4-star rating.
21. **stars_5.0**: Total number of times the restaurant received a 5-star rating.

Attributes related features

The remaining 137 features are attributes, venues, and cuisine related features that consist of binary value.

- **Venues**: Restaurants, Pubs, Cafe, etc.
- **Cuisines**: Indian, Chineses, American (new), Mexican, etc.
- **Attributes**: Takeout, counterservice, delivery, dinner, etc.

5. Statistical Analysis

Utilized three different statistical test methods:

- **ANOVA (Analysis of Variance)** - Testing categorical and quantitative Variables to determine whether is_open and lifespan columns are related.
- **Chi-Square** - Testing two categorical variables.
- **Pearson Correlation** - Testing two quantitative variables

I based my decision whether to reject null hypothesis (H_0) or not by following standard p-value cutoff of 0.05 or 5%.

- p-value < α (0.05) - Data providing significant evidence against the null hypothesis (H_0) and accept the alternate hypothesis (H_a). In other words, it is more than 95% likely that the association of interest would be present following repeated samples drawn from the population.
- p-value > α (0.05) - Data do not provide enough evidence against the null hypothesis (H_0)

ANOVA Testing - Categorical and Quantitative Variables

Evaluated whether or not there's an association between restaurant's current business status (close or open - categorical variable) and how long the restaurant has been operating for (lifespan - quantitative variable). I also evaluated the relationship between the restaurant's business status and its revenue.

Restaurant Status and Lifespan

- H_0 - Restaurant's lifespan and restaurant's business status are ***unrelated***.
- H_a - Restaurant's lifespan and restaurant's business status are ***related***.

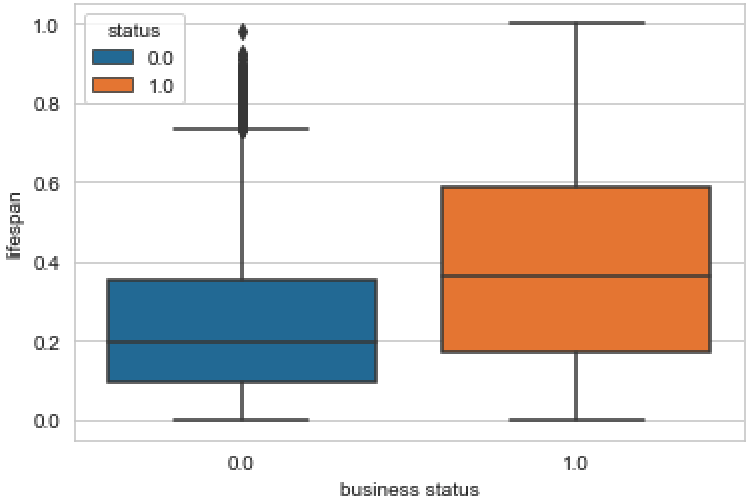
Restaurant Status and Revenue

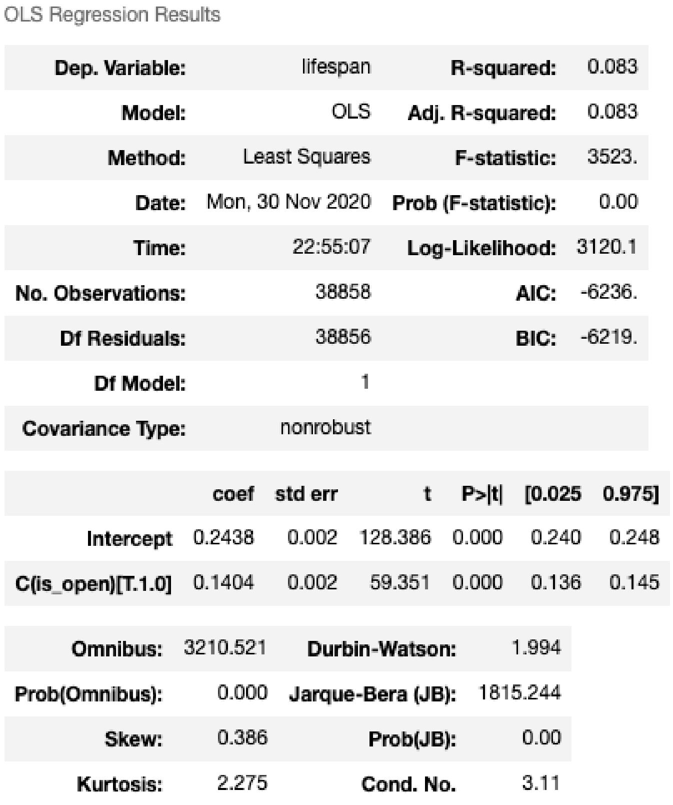
- H_0 - Restaurant's revenues and restaurant's business status are ***unrelated***.
- H_a - Restaurant's revenues and restaurant's business status are ***related***.

Open and close restaurant's snapshot:

- Open restaurants had average lifespan of **0.38** with standard deviation of **0.24** - Restaurants that are *closed for business* has average lifespan of **0.24** with standard deviation of **0.18**
- Restaurants that are *open for business* has average revenue of **0.006** with standard deviation of **0.013**
- Restaurants that are *open for business* has average revenue of **0.003** with standard deviation of **0.007**

While it is true that 0.38 lifespan is more than 0.24 lifespan, it's not at all clear that this is a large enough difference to reject the null hypothesis. Or to say that restaurants with higher lifespan are more likely to be open than restaurants with low lifespan. So we need to assess the evidence in order to determine whether the data provides strong evidence against the null hypothesis to claim that there is no relationship between lifespan and restaurant business status (is_open).



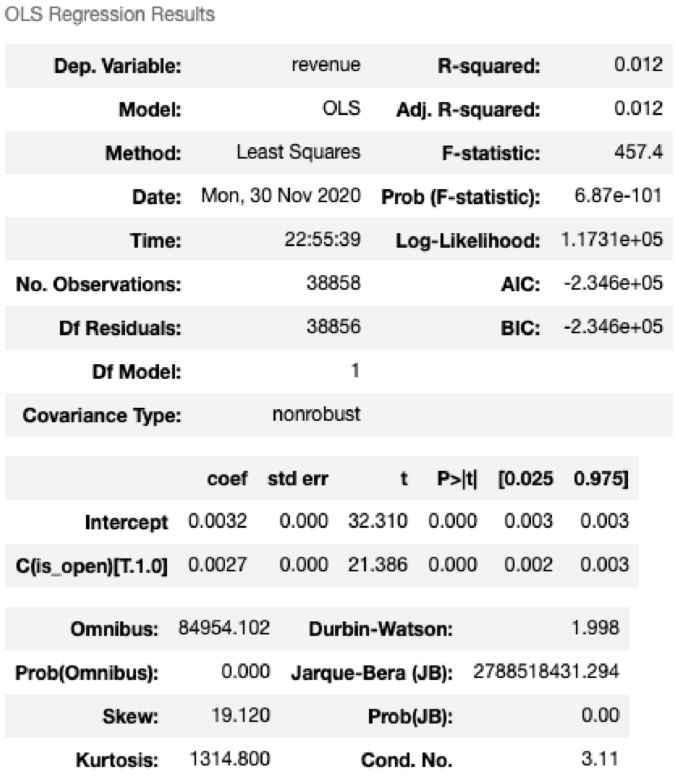


OLS Regression Results

Dep. Variable:	lifespan	**R-squared:**	0.083
Model:	OLS	**Adj. R-squared:**	0.083
Method:	Least Squares	**F-statistic:**	3523.
Date:	Mon, 30 Nov 2020	**Prob (F-statistic):**	0.00
Time:	22:55:07	**Log-Likelihood:**	3120.1
No. Observations:	38858	**AIC:**	-6236.
Df Residuals:	38856	**BIC:**	-6219.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.2438	0.002	128.386	0.000	0.240	0.248
C(is_open)[T.1.0]	0.1404	0.002	59.351	0.000	0.136	0.145

Omnibus:	3210.521	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1815.244
Skew:	0.386	**Prob(JB):**	0.00
Kurtosis:	2.275	**Cond. No.**	3.11



OLS Regression Results

Dep. Variable:	revenue	**R-squared:**	0.012
Model:	OLS	**Adj. R-squared:**	0.012
Method:	Least Squares	**F-statistic:**	457.4
Date:	Mon, 30 Nov 2020	**Prob (F-statistic):**	6.87e-101
Time:	22:55:39	**Log-Likelihood:**	1.1731e+05
No. Observations:	38858	**AIC:**	-2.346e+05
Df Residuals:	38856	**BIC:**	-2.346e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.0032	0.000	32.310	0.000	0.003	0.003
C(is_open)[T.1.0]	0.0027	0.000	21.386	0.000	0.002	0.003

Omnibus:	84954.102	**Durbin-Watson:**	1.998
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2788518431.294
Skew:	19.120	**Prob(JB):**	0.00
Kurtosis:	1314.800	**Cond. No.**	3.11

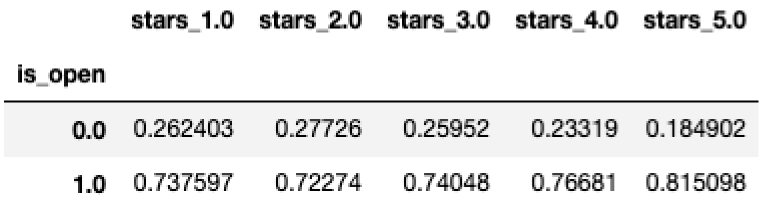
Both Anova tests yielded incredibly small p-value with high F-statistics which tells us that there is an association between restaurant lifespan and business status (is_open) as well as restaurant's revenue and its business status.

Chi Square Test - Two Categorical Variables

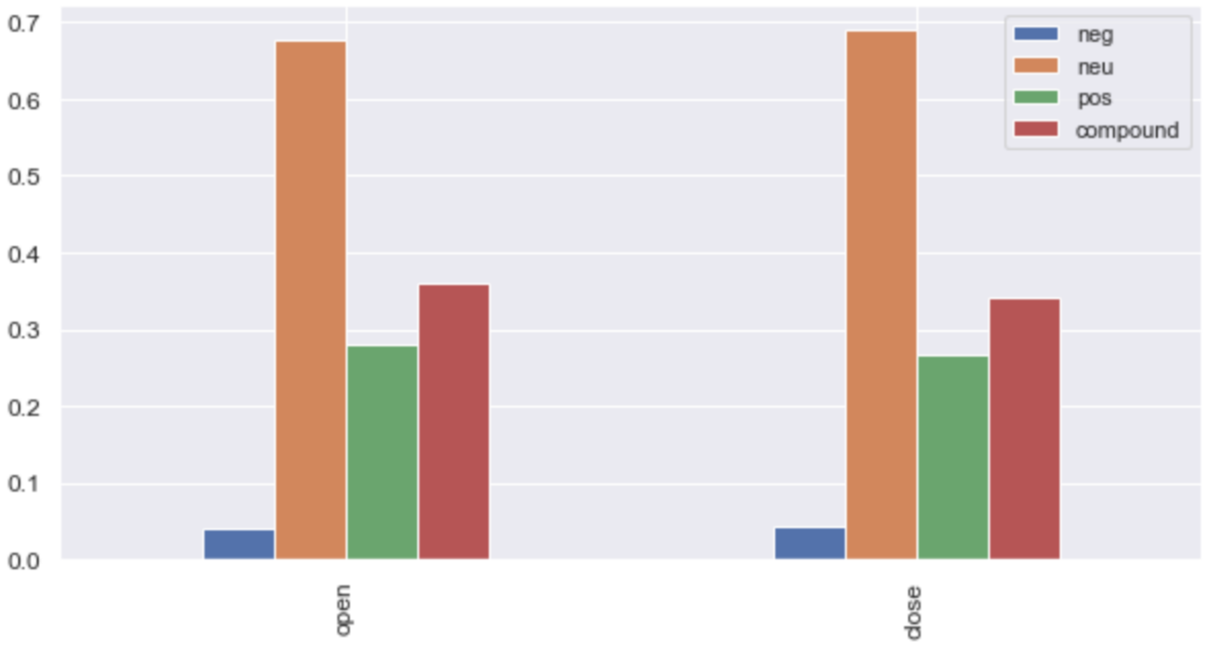
I was curious to see if there was a relationship between restaurant's business status and multiple stars columns (range from 1.0-5.0).

- H_0 - Restaurant's lifespan and restaurant's star rating counts are ***unrelated***.
- H_a - Restaurant's lifespan and restaurant's star rating counts are ***related***.

Initially, I grabbed 5 columns related to stars rating (which has count values per restaurant) and is_open column for testing. Afterwards I created a contingency table which both rows and columns accumulate to 100%.



	stars_1.0	stars_2.0	stars_3.0	stars_4.0	stars_5.0
is_open					
0.0	0.262403	0.27726	0.25952	0.23319	0.184902
1.0	0.737597	0.72274	0.74048	0.76681	0.815098

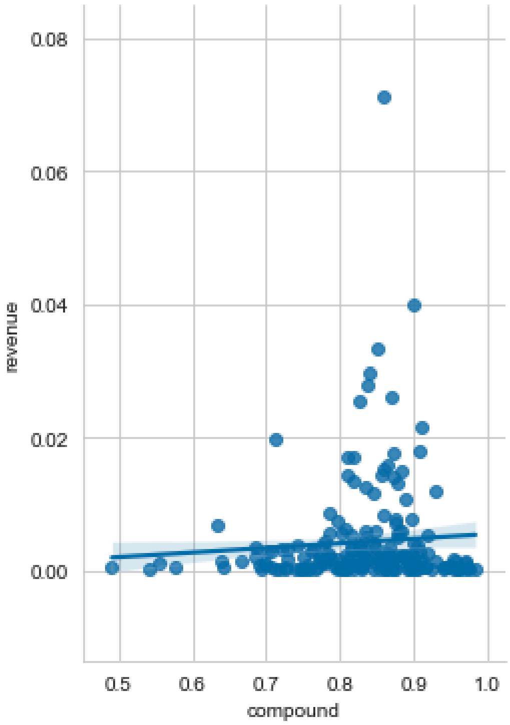


After testing with chi-square, it returned a high chi-square value of **48**, and the p-value shown in scientific notation is quite small, approximately 8.7e to the negative 10, which clearly tells us that stars columns and is_open columns are significantly associated. Another interesting to note is that open restaurants tend to have more positive stars (3.0-5.0) compared to close restaurants.

Pearson Correlation - Two Quantitative Variables

Earlier in this section, we found out that the restaurant's revenues and its business status are associated. In other words, a restaurant's revenues have some influence whether restaurants will remain open or not. I wanted to find whether revenues and sentiment's compound score has a strong positive linear relationship using pearson correlation.

- H_0 - Restaurant's revenue and restaurant's sentiment's compound score are ***unrelated***.
- H_a - Restaurant's revenue and restaurant's sentiment's compound score are ***related***.



Pearson correlation returned correlation coefficient of approximately **0.08** with a very small p-value of 1.7e negative 57. The relationship between revenue and compound are statistically significant with a slight positive linear relationship as shown on the graph above. However, residuals (distance between data points and line) for the most part seem okay as the majority of the data points are congested in one area but there are a significant number of data points that are far from the line.

Summary of Statistical Analysis

- **ANOVA (Analysis of Variance)** - Tested restaurant's lifespan and revenue (explanatory variables) and its business status (response variable) which concluded lifespan/revenue and is_open columns are associated, rejecting null hypothesis due to having very small p-value.
- **Chi-Square** - Tested stars columns (1.0-5.0) and restaurant's lifespan which also resulted in having association, open restaurants had significantly higher ratings count.
- **Pearson Correlation** - Tested between restaurant's revenue and its sentiment's compound score - it has slightly positive linear relationship with low p-value, most data points are congested in one area, however there are significant number of data points that have high residual value.

6. Machine Learning

For this particular classification problem, I decided to use multiple machine learning algorithms that utilizes ensemble learning. Below are the following machine learning algorithms that were used to predict whether a restaurant will fail or remain open with the given features.

1. **Random Forest** (Bagging)
2. **AdaBoost** (use of increasing the weight of misclassified data points)
3. **Gradient Boosting** (learning previous mistakes with residual error)

Based on the result from those three machine learning algorithms, I'll be determining the best machine learning algorithm based on AUC and computational time. All algorithms were analyzed by initially using default parameters and making improvements by optimizing model parameters using either RandomizedSearchCV and GridSearchCV.

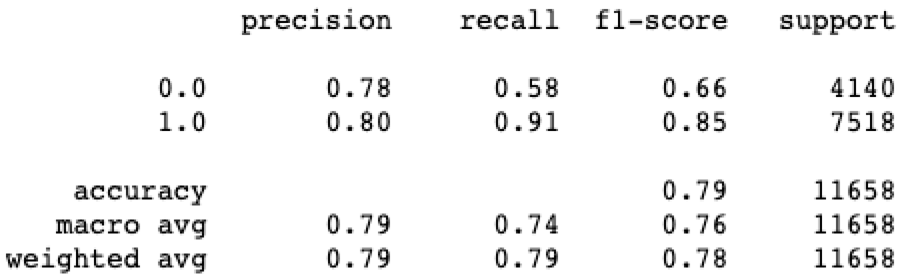
As standard, features were tested using standard train/test split ratio of 7:3. 70% were fitted to the model and 30% were left for testing to evaluate machine learning algorithms. The baseline was having at least 100 decision trees (or stumps for AdaBoost and Gradient Boosting).

6.1 Random Forest (Bagging Approach)

My focus is to prevent or minimize overfitting or having high false positives in the result; therefore, random forest is used to check since it minimizes overfitting and handles large dataset with high dimensionality.

Default approach without hyperparameter tuning

With the initial baseline result with 100 decision trees (n_estimators) it yielded the following result below:



```
              precision    recall  f1-score   support

         0.0       0.78      0.58      0.66      4140
         1.0       0.80      0.91      0.85      7518

    accuracy                           0.79     11658
   macro avg       0.79      0.74      0.76     11658
weighted avg       0.79      0.79      0.78     11658
```

Initial Precision and Recall

According to the initial confusion matrix report, random forest classifier is better at identifying open restaurants compared to close businesses. **42% being false positive** and **8% being false negative**. Since this capstone project is about whether to lend money to the restaurant or whether aspiring restaurateurs should open a restaurant. I need to adjust my machine learning algorithm to focus on reducing false positives because we don't want to lend or invest in businesses that will eventually close.

Selecting the best tuning parameters (aka 'hyperparameters') for Random Forest

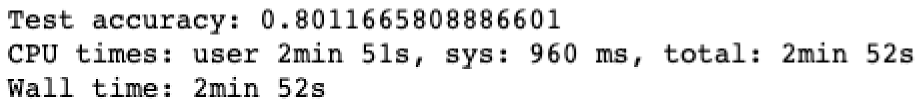
Randomized search cv. Randomized search cv is used for Random Forest due to random forest having many parameters which may take a lot of computational time in finding best parameters without overfitting. Below are the parameters I'll be tuning:

- n_estimators = number of trees in the forest
- max_features = max number of features considered for splitting a node
- max_depth = max number of levels in each decision tree
- min_samples_split = min number of data points placed in a node before the node is split
- min_samples_leaf = min number of data points allowed in a leaf node
- bootstrap = method for sampling data points (with or without replacement)

Randomized Search CV yielded the best parameters:

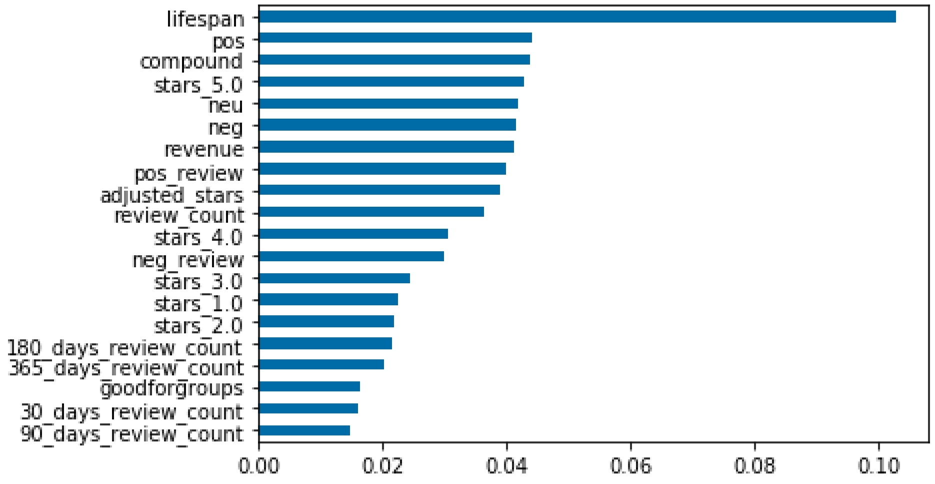
- n_estimators: 1800
- min_samples_split: 2
- min_samples_leaf: 1
- max_features: 'auto'
- max_depth: 50
- criterion: entropy
- bootstrap: False

Test result



```
Test accuracy: 0.8011665808886601
CPU times: user 2min 51s, sys: 960 ms, total: 2min 52s
Wall time: 2min 52s
```

Feature Importance



RandomForest Summary

Randomized Search CV improved its overall AUC by 1% by improving in identifying true negatives but it did not significantly reduce false positives. Random Forest gave equal weights to all created decision trees which resulted in about 80% AUC; we will identify whether giving different weights to each decision tree will give better results by using Adaptive Boosting.

Lifespan, pos, and star rating, and sentiment related features had the highest signal in determining whether a restaurant will fail or not. With a strong emphasis on lifespan features. We will later identify whether this is true for the other two algorithms we will see in the later section.

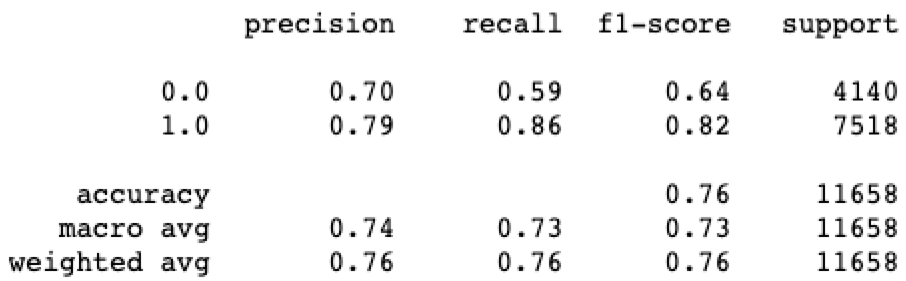
6.2 AdaBoost (Sampling Distribution)

I decided to use the AdaBoost algorithm to inspect how well it performs when we focus on what the model misclassified through **sampling distribution**. In other words, we give more focus on data points that make a mistake so that it learns from its mistake when creating n-stumps.

Default approach without hyperparameter tuning

Inspecting baseline when AdaBoost is used without tuning with the exception of n_estimators (how many stumps we like to create).

Initial Recall and Precision



```
              precision    recall  f1-score   support

         0.0       0.70      0.59      0.64      4140
         1.0       0.79      0.86      0.82      7518

    accuracy                           0.76     11658
   macro avg       0.74      0.73      0.73     11658
weighted avg       0.76      0.76      0.76     11658
```

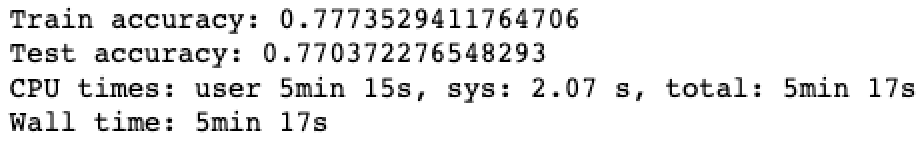
AdaBoost algorithm took **5.45s** in creating and evaluating 100 decision trees with **76% AUC** without hyperparameter tuning. Overall AdaBoost provided less desirable results compared to random forest classifiers since AdaBoost had higher **false negatives at 14%**. However, it did provide lower **false positives at 40.5%, decrease from 42%**.

Selecting the best tuning parameters for AdaBoost

Will be using GridSearchCV for optimization by tuning the following parameters below:

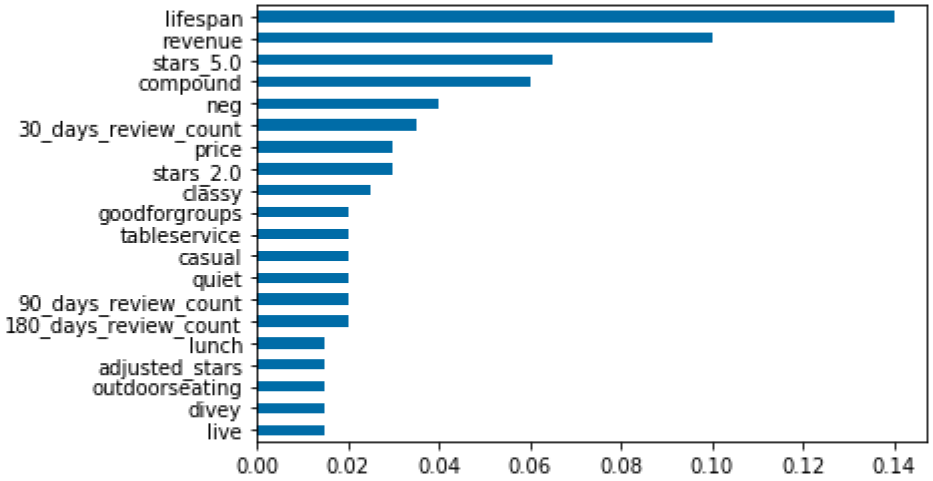
- **n_estimators**: maximum number of estimators (stumps at which boosting is terminated
- **learning_rate**: rate at which we are adjusting the weights of our model with respect to the loss gradient

Test result



```
Train accuracy: 0.7773529411764706
Test accuracy: 0.770372276548293
CPU times: user 5min 15s, sys: 2.07 s, total: 5min 17s
Wall time: 5min 17s
```

Feature Importance



AdaBoost's feature importance is slightly different with the last 30 day review count and several other restaurant attributes turned out to be more important in identifying closed and open restaurants. Lifespan, revenue, and sentiment score related features are consistently ranked high in determining our restaurant's business status.

AdaBoost Summary

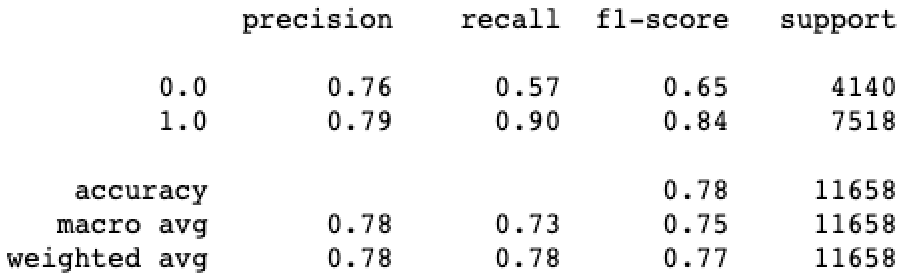
Initial AdaBoost algorithm with default setting took **5.45 s** in creating and evaluating 100 decision trees with **76%** AUC which is lower than Random Forest's initial AUC. However it did provide lower false positives at **40.5%**, decrease from **42%.** I tuned its hyperparameters using GridSearchCV adjusting **n_estimators at 200** with **learning_rate of 0.5**.

It yielded a better overall result at **77%** getting better results at obtaining higher true negatives but at the expense of gaining more false positives.

6.3 Gradient Boosting

AdaBoost provided a fairly okay result in identifying closed and open restaurants through sampling distribution and giving each stump its own weight in deciding which model worked best. I decided to use gradient boosting to see if it provides a similar or better result to Random Forest (which currently holds best AUC) through residual error directly instead of giving weights to each data point.

Initial Recall and Precision



```
              precision    recall  f1-score   support

         0.0       0.76      0.57      0.65      4140
         1.0       0.79      0.90      0.84      7518

    accuracy                           0.78     11658
   macro avg       0.78      0.73      0.75     11658
weighted avg       0.78      0.78      0.77     11658
```

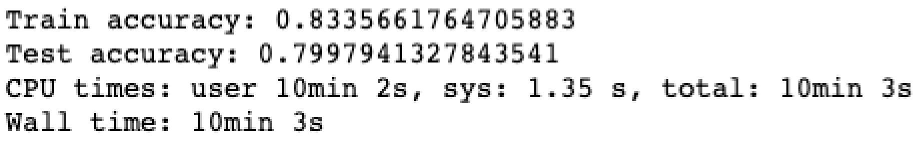
Gradient boosting did better at predicting open and closed restaurants than AdaBoost but did poorer than Random Forest, similar to all algorithms - It did not do a very good job at predicting closed restaurants - instead it misclassified closed restaurants as open. It has the highest false positives (**43%**) compared to other algorithms. Overall it has initial AUC at **78%**.

Selecting the best tuning parameters for Gradient Boosting

Same as AdaBoost, I will be using GridSearchCV for optimization by tuning the following parameters below:

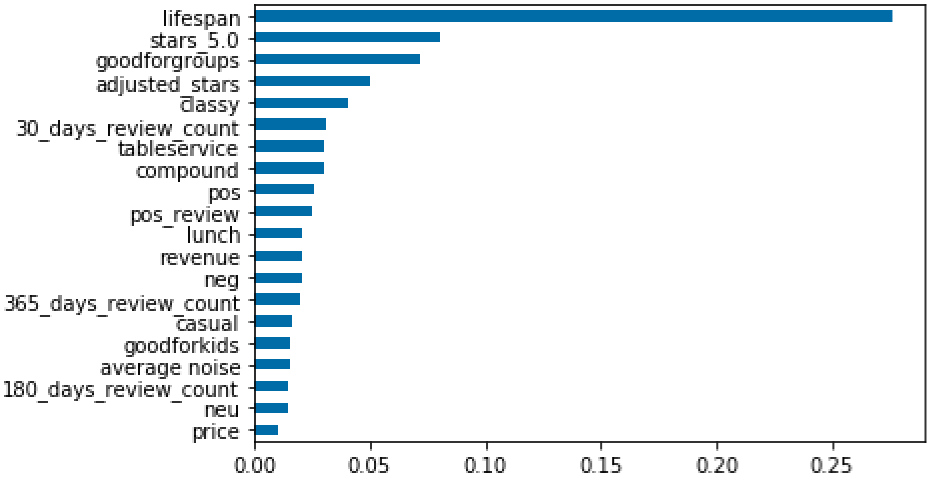
- **n_estimators**: maximum number of estimators (stumps at which boosting is terminated
- **learning_rate**: rate at which we are adjusting the weights of our model with respect to the loss gradient

Test result



```
Train accuracy: 0.8335661764705883
Test accuracy: 0.7997941327843541
CPU times: user 10min 2s, sys: 1.35 s, total: 10min 3s
Wall time: 10min 3s
```

Feature Importance



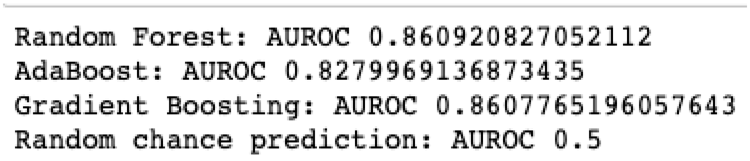
Gradient Boosting Summary

Initial GradientBoosting algorithm with default setting took **9.24 s** in creating and evaluating 100 decision trees with **78%** AUC which is lower than Random Forest's initial AUC but higher than AdaBoost's AUC. However it did provide the highest false positives at **43%**. I tuned its hyperparameters using GridSearchCV adjusting **n_estimators at 200** with **learning_rate of 0.2**.

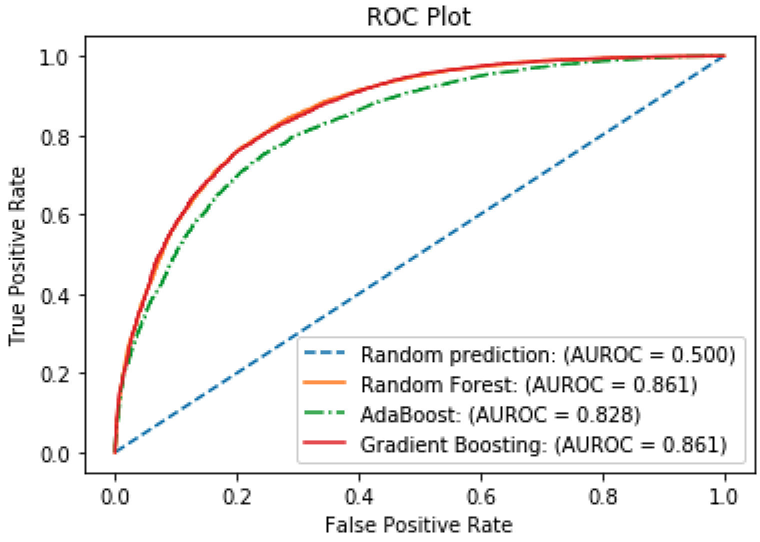
It yielded the best overall AUC result at **79.9%** and getting least false positives at **36.9%**. So far, I would recommend using gradient boosting as it has a lower chance of lending or investing on businesses that are going to fail.

6.4 ROC and AUC

Computing AUC and ROC curve values and plotting for visualization purpose.



```
Random Forest: AUROC 0.860920827052112
AdaBoost: AUROC 0.8279969136873435
Gradient Boosting: AUROC 0.8607765196057643
Random chance prediction: AUROC 0.5
```



Summary

Random forest and Gradient Boosting algorithms gave the best outcome in predicting whether restaurants are open or closed based on given features. Each algorithm had differing feature importance but few features repeatedly came into view such as lifespan, sentiment score, star rating, revenue, and review count. Those five features played an important role in determining whether a restaurant will strive or fail in the hospitality industry with precision of **84%** and overall F1-score of **80%.**

7. Suggested Improvement

There are several ways to improve performances in identifying a restaurant's business status using multiple data sources.

- Using population demographics and income level to gauge if it impacts a restaurant's price range or its revenues.
- Comparing nearby similar competitors - whether one similar restaurant's performance affects nearby restaurant's performance.
- Using actual restaurant's revenue data instead of using speculative revenues which were calculated using price and review_count columns.

8. Project Summary

- The model was built for restaurant lending purposes that helps decide whether to invest in an independent restaurant or not based on its given data.
- Yelp dataset was used and analyzed to build a classification model that correctly identifies restaurant's business status.
- Four major predictive features were identified which is lifespan, sentiment analysis scores, star ratings, and revenues.
- Used three ML algorithms based on bagging and boosting (Random Forest, AdaBoost, and Gradient Boosting) which yielded the highest precision score of **79%** and recall score of **90%** with overall F1-score of **80%.**

None of the restaurant's attributes such as cuisines, service types, and venue types yielded any significant result in identifying the restaurant's status. Price (general dining cost) did not matter whether the restaurant will remain open or not as indicated in Machine Learning Algorithm's feature importance. In conclusion, As common as this may sound, having long lifespan and positive sentiment scores were shown to be great predictors in identifying healthy stable restaurants as proven through data analysis and multiple machine learning algorithms.