DSA 5900 Professional Practice

Department of Data Science and Analytics

Spring 2021

Predicting Business Owners Response on Social Media Using Data Science Approach

Purushotham Vadde – 113459984

Supervisor:

Naveen Kumar,

Assistant Professor of Management Information Systems,

University of Oklahoma

Table of Content

1. Introduction

2. Objectives

3. Preparing Dataset

   3.1 Data source
   3.2 Data collection
   3.3 Data description
   3.4 Data understanding
   3.5 Data preparation
   3.6 Data transformation
   3.7 Feature engineering

4. Methodology

   4.1 Random forest
   4.2 Gradient boosting
   4.3 XGBoost
   4.4 Important features
   4.5 Oversampling with SMOTE

5. Results and Analysis

6. Deliverables

7. Self- Assessment

8. References

1. Introduction:

Nowadays, most online users rely on online reviews of a business or product before making a decision. Online reviews are playing a central role in decision-making. According to Forbes, 90% of the user will read the reviews of a business to determine the quality, 82% of Yelp users said they typically visit Yelp because they intend to buy a product or service. 74% of users say that the positive reviews make them trust a local business more[1]. The Internet is filled with many reviews, and there are thousands of reviews available for each business. People prefer a business with good reviews compare to the other. All Web applications allow users to write a review of their experience with their respective businesses to know about their service. There are many applications, especially for business reviews, such as Yelp, Google reviews, Facebook, Etc.

The online reviews contain the reviews from customers and business owners' responses for the customer reviews. We can also see the customer reviews are more in number compared to business owner response reviews. The reason for this is that the business owners are not responding to all the customer reviews. They are responding to only a few reviews.

As part of this project, we will develop a Data science approach to determine what causes the business owner to respond to a review about their business or product. We will use the supervised machine learning approach called classification to find the type of reviews for which the Business owner will respond. We are mainly going to use ensemble methods such as Bagging and Boosting algorithms to classify business reviews.

2. Objective:

The main goal of this project is to build a Binary Classification machine learning model using ensemble algorithms to classify the online business reviews for which the Business Owners are likely going to respond. Another critical step in this project is data collection, the business reviews for different business categories to be collected from the Yelp website through web scraping. We will also focus on finding the key insights, such as the features that affect business owners to respond to the customer review using descriptive Analytics.

We will also work on model performance tuning approaches to create a model that produces the best results. We will work on model performance evaluation using the evaluation metrics correct for our model and data set.

3. Preparing dataset

3.1 Data source:

For this project we have taken the Yelp as the Data source for collecting the reviews. Yelp is a social media network where everyone can create an account and share their experiences with any of the businesses listed on the yelp website. There are over 1200 different business categories on Yelp, and the list is still rising[1]. An individual with a Yelp account can write a review for any of the 1200 businesses listed on the site. Other Yelp users can see the reviews left by other users for any of the businesses listed. In Yelp, we can also see the Business owner replies to customer feedback for each business. We focused on Seattle for this project, as well as eight categories that are closely connecting to people's daily lives (Auto Repair, Flooring, Handyman, HVAC, Painters, Plumbing, Tires, Transmission Repair)

3.2 Data collection:

3.2.1 Available data:

On the Yelp website, we have millions of reviews, and more are adding every day in each business, but these reviews are not available for public study. For our project, the Review data available in the Yelp Developers community is inadequate. As a result, we do not have the data we need to target specific locations and categories for our project. So, we will have to figure out how to gather data on our own for our project.

3.2.2 Challenges in collecting new data:

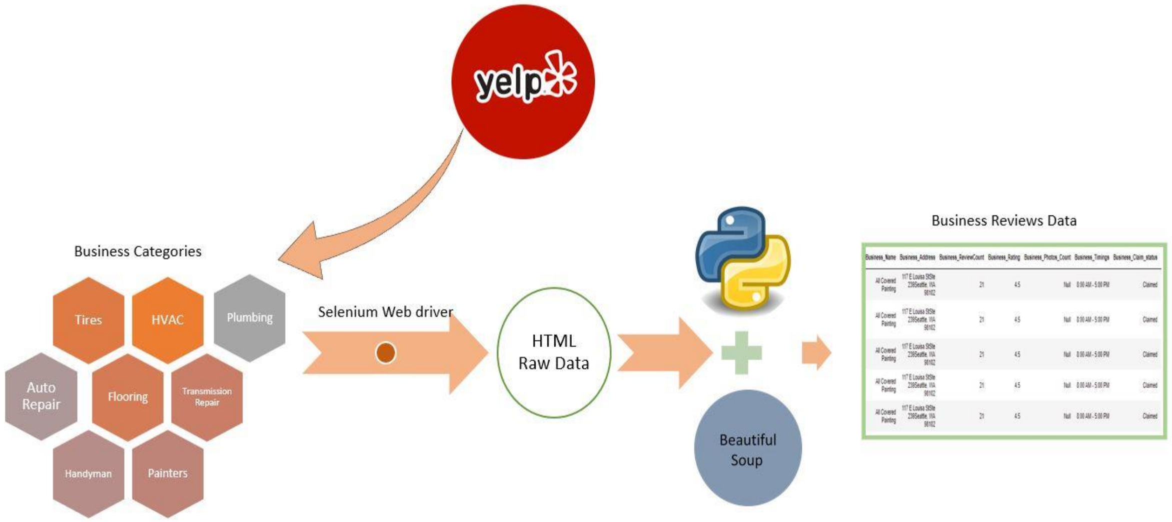
To collect data from Yelp, the company provides an API from which we can obtain business reviews, but we can only obtain a maximum of three business reviews. Scraping the Yelp website was the only way to get the business feedback that we needed.

Web scraping the Yelp website is difficult due to the way it is designed. To extract the html content of web pages, we typically use the urllib and requests python packages. Only after the website is loaded with the business URL will we get the business reviews data in HTML. So, by simply calling the business URL with the packages urllib and requests, we will not be able to get the complete business reviews HTML data.

The yelp search results are made with repeated or duplicated businesses to restrict scraping. Yelp only shows the results of up to 240 companies in a search result. So, if we check for any city and category combination, we will not get more than 240 business results. Aside from these difficulties, the HTML tags for each feature on a webpage are complex and change frequently. As a result, maintaining the code base is difficult.

3.2.3 Data collection approach:

As previously mentioned, data collection through web scraping poses a lot of technical challenges. We used the Data collection architecture shown in Fig 1 to solve these challenges. We used the selenium web driver to extract the HTML raw data and the beautiful soup to collect the business reviews data from the HTML, as shown in Fig 1.



Business_Name	Business_Address	Business_ReviewCount	Business_Rating	Business_Photos_Count	Business_Timings	Business_Claim_status
All Covered Painting	117 E Louisa StSte 239Seattle, WA 98102	21	4.5	Null	8:00 AM - 5:00 PM	Claimed
All Covered Painting	117 E Louisa StSte 239Seattle, WA 98102	21	4.5	Null	8:00 AM - 5:00 PM	Claimed
All Covered Painting	117 E Louisa StSte 239Seattle, WA 98102	21	4.5	Null	8:00 AM - 5:00 PM	Claimed
All Covered Painting	117 E Louisa StSte 239Seattle, WA 98102	21	4.5	Null	8:00 AM - 5:00 PM	Claimed
All Covered Painting	117 E Louisa StSte 239Seattle, WA 98102	21	4.5	Null	8:00 AM - 5:00 PM	Claimed

Fig 1: Data collection architecture

Selenium:

Selenium web driver is a framework used to test web applications. Using the selenium web driver and chrome driver, we can extract the HTML content of the web pages. The main advantage of using the selenium web driver is that the selenium can handle dynamic web content. The way selenium works is that it creates a drive object for the Chrome browser and then opens the business web page in the Chrome browser, where we can get the HTML data of the business web page using Chrome's built-in functions. In comparison, the packages such as urllib and requests selenium work effectively for dynamic web content. We solved the problem of dynamic web content and extracted HTML data from web pages using Selenium.

Beautiful soup:

The Beautiful soup is a Python library used to extract data from HTML. The HTML raw data extracted from the Yelp Website parsed using the beautiful soup, and the required Business Reviews data extracted from the HTML page using the Beautiful soup inbuilt functions. and to get the most data, we look for businesses based on zip codes in Seattle, so we can get the most business data. We also maintained a separate HTML tags file so that code maintenance would be simple if the HTML tags changed in the future.

3.3 Data description:

Our final Data set consists of Business reviews from Seattle City in eight different categories such as (Auto Repair, Flooring, Handyman, HVAC, Painters, Plumbing, Tires, Transmission Repair). A total of 97584 customer reviews have been collected. We have collected the 24 features for each business review. Please see Appendix 1 for a complete list of business features that we have collected for each review. These features are collected in 3 levels, such as business-related features, Customer-related features, and business owners' responses.

Based on the data we collected, we found that 17804 customer reviews received a business response of 18.24 percent, and 79780 customer reviews received no business response of 81.75 percent we can see the same data in the Fig 2.

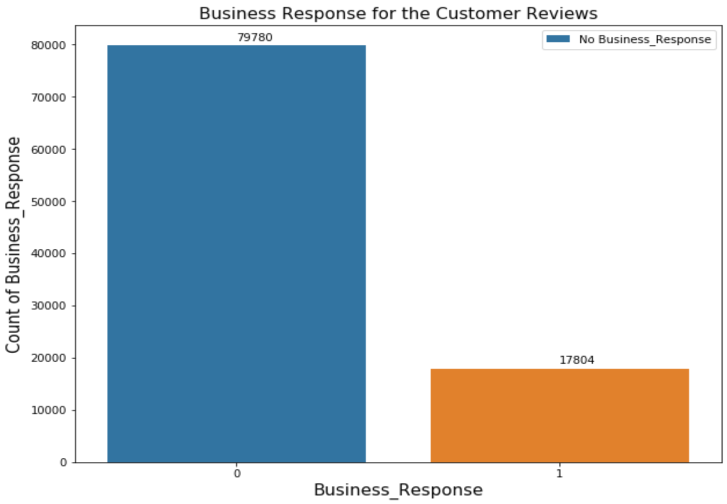
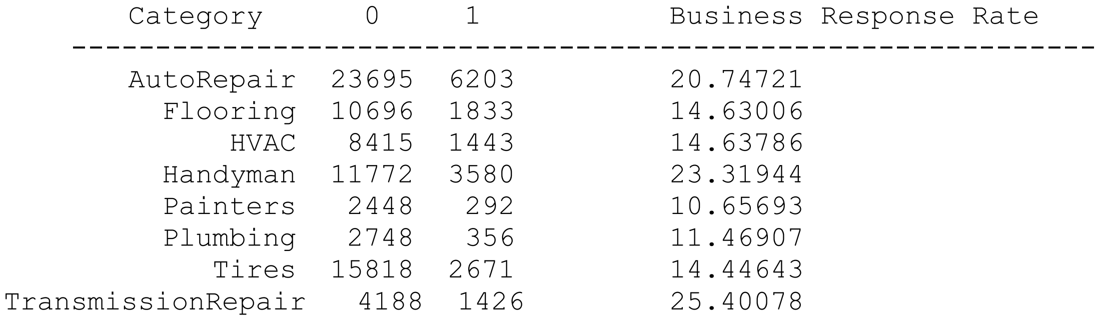


Fig 2: Total Reviews VS Business Response



```
    Category      0     1          Business Response Rate
--------------------------------------------------------------
    AutoRepair  23695  6203        20.74721
      Flooring  10696  1833        14.63006
          HVAC   8415  1443        14.63786
      Handyman  11772  3580        23.31944
      Painters   2448   292        10.65693
      Plumbing   2748   356        11.46907
         Tires  15818  2671        14.44643
TransmissionRepair   4188  1426        25.40078
```

Table 1 : Business Response category wise

Table 1 shows the business response rate for each category based on the collected reviews. We can see that the transmission repair, handyman, and Auto repair categories have the high business response rate. The explanation for this may be that business owners respond more actively to customer reviews than other businesses, and customers use Yelp more often for these categories in above categories.

3.4 Data understanding:

3.4.1 Business review's count:

In general, businesses that are active on social media are more likely to receive more reviews. Additionally, businesses that are active on social media have a higher response rate for customer reviews. From our dataset we observe the similar behavior where the businesses with more reviews count are more likely to respond to customer reviews than businesses with fewer reviews.

The Mean and Median of business review counts are high for businesses that respond to customer reviews more frequently, as shown in Fig 3. We can see that those actively responding have a median of 104 reviews and a mean of close to 200. So, as shown in Fig 3, the Business Response Rate is higher for businesses with more reviews than for businesses with fewer reviews.

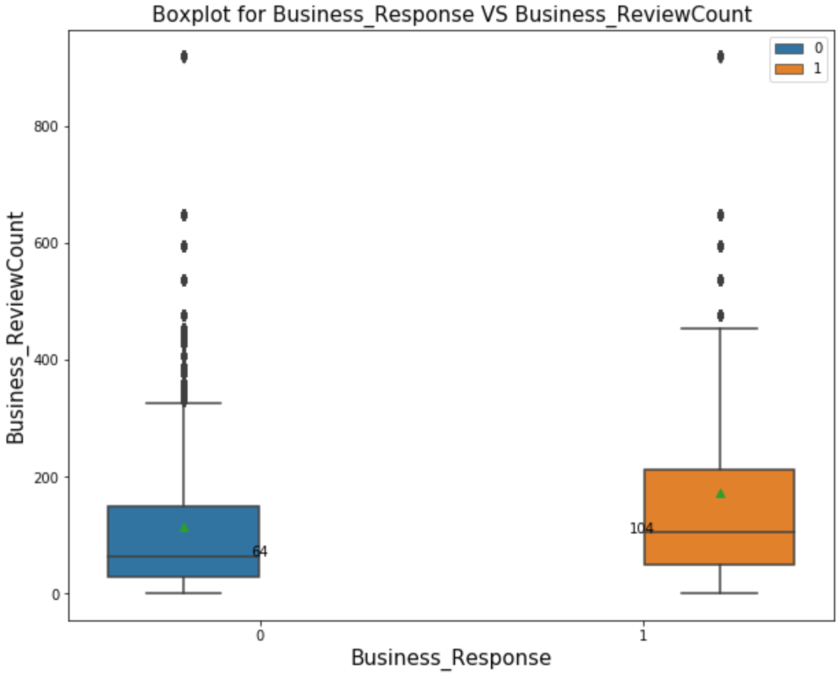


Fig 3: Business Response VS Business Review Count

3.4.2 Business rating:

74% of users say that the positive reviews make them trust a local business more[2]. Business ratings are entirely dependent on positive feedback; as the number of positive reviews for a Business grows, so do the Business ratings. We can see in Fig 4 a boxplot for business rating based on the business response below. We observe that the median rating is the same for reviews that received a response and not receive a response. Whereas the mean is low for the business responded class as the two classed sample size is different, we cannot conclude the mean for the business responded class is low.

Hypothesis testing for two population mean:

We can do the Hypothesis testing to find the two-population means are same or not.

Null Hypothesis = Means are same

Alternative Hypothesis = means are not same

Class0 sample size n_1=79780

Class1 sample size n_2= 17804

Mean of class0 u_1= 4

Mean of class1 u_2 = 3.8

Standard Deviation of class 0 s_1= 0.8757

Standard Deviation of class 1 s_2 = 0.7637

With size, mean and standard deviation of the above two classes and we got the t0 statistic as 30.7253. When we Calculated the p-value based on the t0 statistic with 95 percent confidence, we get a p.value of 0.00001, which is less than 0.05.

As a result, we can reject the null hypothesis that both means are equal and conclude that the mean for the business responding class is lower, implying that businesses with low ratings respond more than businesses with high ratings.

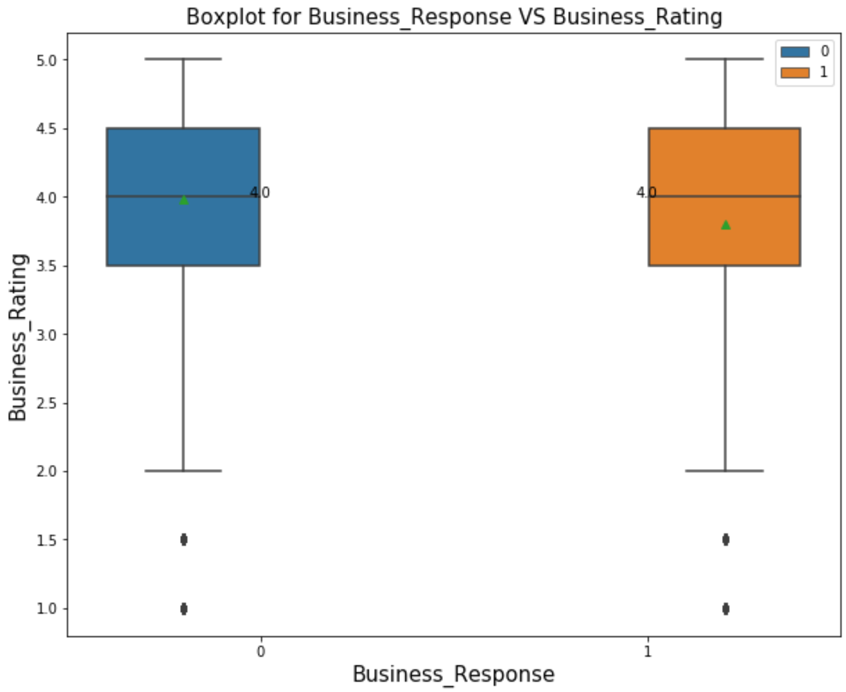


Fig 4: Business Response VS Business Rating

3.4.3 Customer rating:

The customer rating reflects the customer's opinion of the business. Whether a customer is satisfied with a company's service, the rating will be favorable and high; if the customer is dissatisfied with the service, he will give the business a low rating and a negative review. We can see in Fig 5 that the business response rate increases as the customer rating score decreases and decreases as the rating increases, implying that customers with lower ratings are more likely to receive a business response. The business response rate for Rating 1 is 25.31 percent whereas for rating 5 is 15.72 percent.

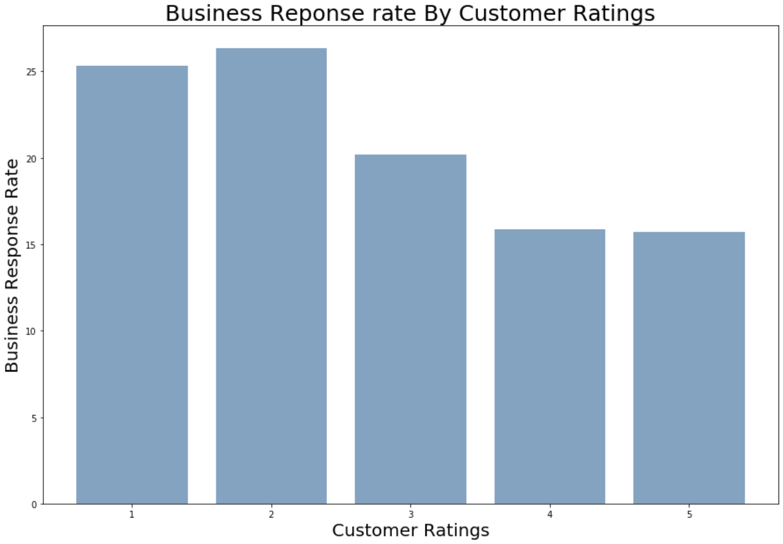


Fig 5: Business response rate based on Customer review rating.

3.5 Feature engineering:

3.5.1 Customer review :

The Customer Review feature represents the review given by the customer for the business. We have a total of 97584 reviews in a dataset. To capture the emotion of the customer, we performed the Sentiment Analysis on the customer reviews.

3.5.2 Review sentiment analysis with TextBlob:

We used Textblob to understand the sentiment of customer reviews. TextBlob is a simple library that supports complex analysis and operations on textual data. The sentiment function in the TextBlob takes the Customer Review as input and returns the two values called Polarity and Subjectivity scores.

Polarity: The polarity value lies between the [-1,1], -1 defines the strong Negative sentiment, and 1 defines the strong Positive sentiment.

Subjectivity: Subjectivity refers to the amount of personal opinion, emotion, or judgment.

Sentiment score: Sentiment Score feature is created based on the polarity score. The Polarity score with greater than zero has set to Positive sentiment reviews. Equal to zero has set to Neutral, and less than zero has set to Negative reviews.

```
Sentiment_Score      0      1        Business Response Rate
------------------------------------------------------------
      Negative   8775   2512                22.25569
       Neutral    653    171                20.75243
      Positive  70352  15121                17.69097
```

Table 2: Sentiment Score VS Business Response

From the Table 2, we can see that the Business response rate is more for negative reviews. We observed the same behavior in the Customer_Reviews_Rating feature, where the Business response rate is high for low rating reviews. So, we can say that we captured the sentiment of customer reviews correctly.

3.5.3 Customer Review statistical features:

When consumers write negative reviews, they tend to use more words to describe their experience than positive reviews. So, we have created few features based on the statistics of customer review such a count of characters in the review, count of words in the review, count of sentences in the review.

3.6 Data transformation:

After the data preparation, we now have a dataset with features that we can use in building the classification model. However, to use features in a machine learning model, they must be in numerical format. Since we have features that include text data, we must convert them to numerical form.

3.6.1 Label encoder: To transform categorical text features into the numerical format, we used the Label encoder from sci-kit learn. The Label encoder takes categorical text data as input and assigns a numerical label to each text category.

3.6.2 Min-Max scaler: The features measured on different scales do not contribute equally to model fitting; thus, the features measured on different scales should be normalized using the min-max scaler.

4. Methodology:

We will use the Ensemble algorithms such as Random Forest, Gradient Boost, Ada Boost, and XGBoost algorithms to build the classification model. Ensemble methods are a machine learning technique that combines multiple base models to create one optimal predictive model.

We need to identify the review data for which the business owner is most likely to respond. It is a binary classification problem, and the data set we are working with is a highly imbalanced dataset with an 18% business response rate. For such dataset, the ensemble classification algorithms work well.

4.1 Random forest:

Random Forest Classifier consists of many decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction, and the class with the most votes become our Random Forest model prediction.

4.1.1 Random forest hyperparameters[4]:

A hyperparameter is a model parameter that helps to improve the model's predictive ability. For our model, we used the following parameters.

max_depth:

In a random forest, the max depth parameter represents the tree's depth, from the longest path root node to the last leaf node.

max_features:

The max_features parameter specifies the maximum number of features that each tree can have. There are many ways to assign max_features, including:

Auto: This will consider all the features that are available in each tree.

Sqrt: for each tree, it will take the square root of the total number of features.

min_sample_leaf:

It represents the minimum number of leaves present in each node before splitting; if the minimum defined samples are not present, the tree will not grow any further, which will help prevent model overfitting.

min_samples_split:

It tells the tree in a random forest how many observations are needed in each node for it to split. By adjusting the min_samples_split feature we can avoid the model over fitting.

n_estimators:

It represents the total number of trees in the model; as the number of trees grows, it increases the model's performance; however, as the number of trees grows, it increases the model's time complexity.

4.1.2 Performance tuning with GridsearchCV:

Trying several parameter combinations manually to find the optimal parameter combination that produces the best result is complicated. GridsearchCV can help in the search for the best performing parameters in the parameter space for a particular algorithm and data set. It runs through all the different parameters fed into the parameter grid and produces the best combination of parameters based on a scoring metric of your choice (f1, recall, accuracy).

4.1.3 Random forest model performance:

Using the GridsearchCV and the scoring metric as accuracy, we got the best model performance results as follows model accuracy as 0.86, and for class 1, we got recall value as 0.40 and F1 score as 0.53.

Even though our model accuracy is 0.86, we can see the low F1 score and Recall because our dataset is unbalanced. To evaluate Model results, we can plot the AUPR Curve and observe model Performance.

From Fig 6 we can see the plots for AU ROC curve, AU PR Curve, F1-score VS Threshold curve.

In the case of an unbalanced data collection, we must consider the AU PR Curve, a precision vs recall curve. In plot 2, the region under the curve for Precision and Recall is 0.6399. We can also see in plot 3 that keeping the predicting threshold at 0.27 results in the highest f1 score of 0.613.

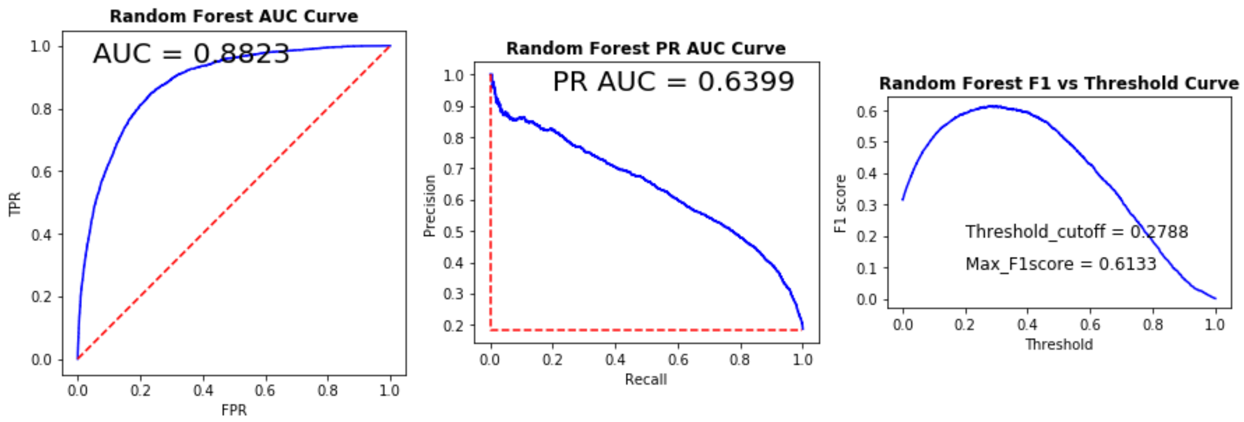


Fig 6 Performance Metric Plots for Random Forest

4.2 Gradient boosting:

Gradient Boosting is an ensemble machine learning algorithm that combines the weakest learners to improve prediction accuracy. Any tree T in the model is constructed based on the model's previous tree T-1 results[5]. The tree outcomes that correctly predicted are given a lower weight, while those incorrectly classified are given a higher weight. A new data set is built based on the weights, and predictions are made. This process repeats for many iterations, all trees are given a weight based on their accuracy, and a consolidated result is generated.

4.2.1 Gradient boosting parameters:

We used the below hyper parameters for tuning the Gradient Boosting model.

max_depth: The max depth parameter represents the tree's depth, from the longest path root node to the last leaf node. The model tends to overfit as the depth of the tree grows.

n_estimators: It represents the total number of trees in the model; as the number of trees grows, it increases the model's performance; however, as the number of trees grows, it increases the model's time complexity.

min_samples_split: It tells the tree in a random forest how many observations are needed in each node for it to split. By adjusting the min_samples_split feature we can avoid the model over fitting.

learning_rate: The learning rate is the time it takes for an error to be corrected from one tree to the next. The model corrects the prediction error faster as the learning rate increases, and the model corrects the prediction error at a slower rate as the learning rate decreases.

4.2.2 Gradient boosting model performance:

Using GridsearchCV and cross-fold validation as ten and scoring metric as accuracy, we obtained the performance results shown for the best model. The model accuracy was 0.85, and the recall value was 0.41, and the f1 score was 0.50 for class 1.

We need to see the PR curve to evaluate the model trained with an imbalanced data set, as we discussed earlier. We can see that the area under the curve for PR Curve is 0.57, and we can also see in plot 3 in Fig 7 that the maximum f1 score for the Gradient Boosting model is 0.56 at the threshold cutoff of 0.019.

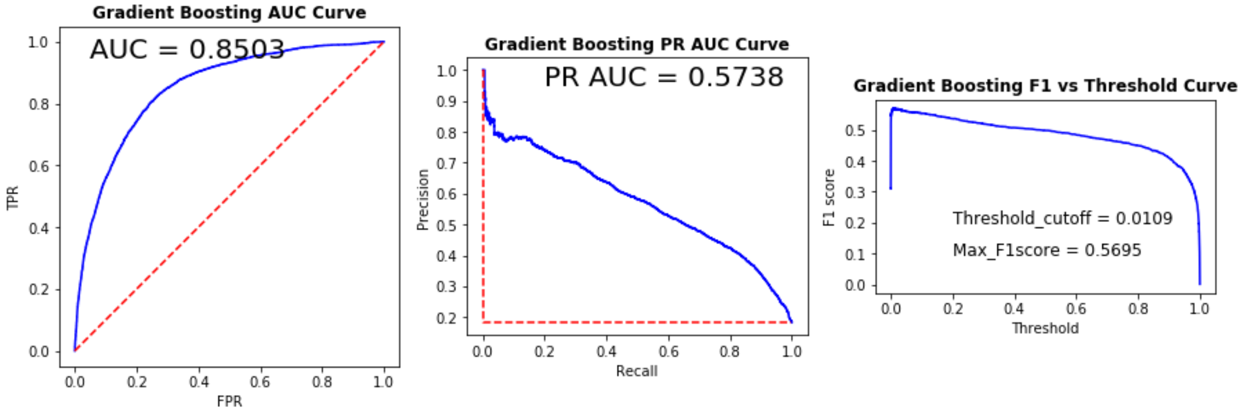


Fig 7: Performance Metric Plots for Gradient Boosting

4.3 XGBoost :

XGBoost has similar behavior to a decision tree in that each tree is split based on a specific range of values in different columns, but unlike decision trees, each node is given a weight. On each iteration a new tree is created, and new node weights are assigned based on similarity score and gain. For each tree, the training examples with the biggest error from the previous tree are given extra attention so that the next tree will optimize more for these training examples, this is the boosting part of the algorithm[6].

4.3.1 Tuning parameters of XGBoost:

colsample_bytree: The number of columns that must be considered in the model is represented by this parameter. The range of values is 0 to 1. If colsample_bytree = 0.5, half of the columns are considered when constructing a tree.

max_depth: The max depth parameter represents the tree's depth, from the longest path root node to the last leaf node. The model tends to overfit as the depth of the tree grows.

min_child_weight: min_child_weight is minimum sum of instance weight needed in a child; The tree won't grow if the tree partition step results in a leaf node with a total of instance weight less than min child weight.

learning_rate: The learning rate is the time it takes for an error to be corrected from one tree to the next. The model corrects the prediction error faster as the learning rate increases, and the model corrects the prediction error at a slower rate as the learning rate decreases.

4.3.2 XGBoost model performance:

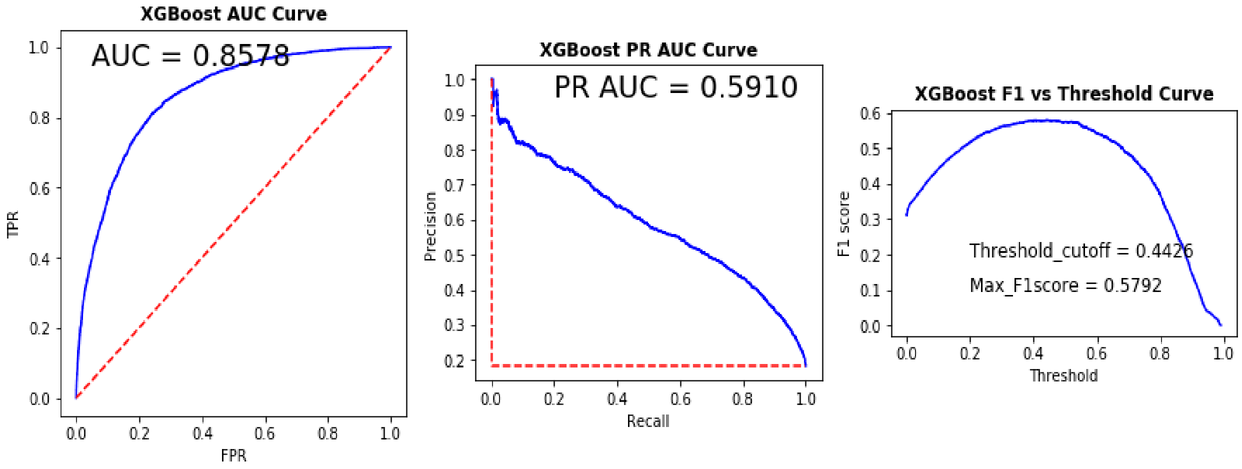


Fig 8: XGBoost Model Performance

Using the GridsearchCV and the accuracy scoring metric, we achieved the best model performa nce results, with model accuracy of 0.83, recall of 0.63, and f1 score of 0.57 for class 1.
We need to see the PR curve to evaluate the model trained with an imbalanced data set, as we discussed earlier. We can see that the area under the curve for PR Curve is 0.59, and we can als o see in plot 3, in Fig 8 that the maximum f1 score for the XGBoost model is 0.579 at the thresh old cutoff of 0.4426.

4.4 Important features:

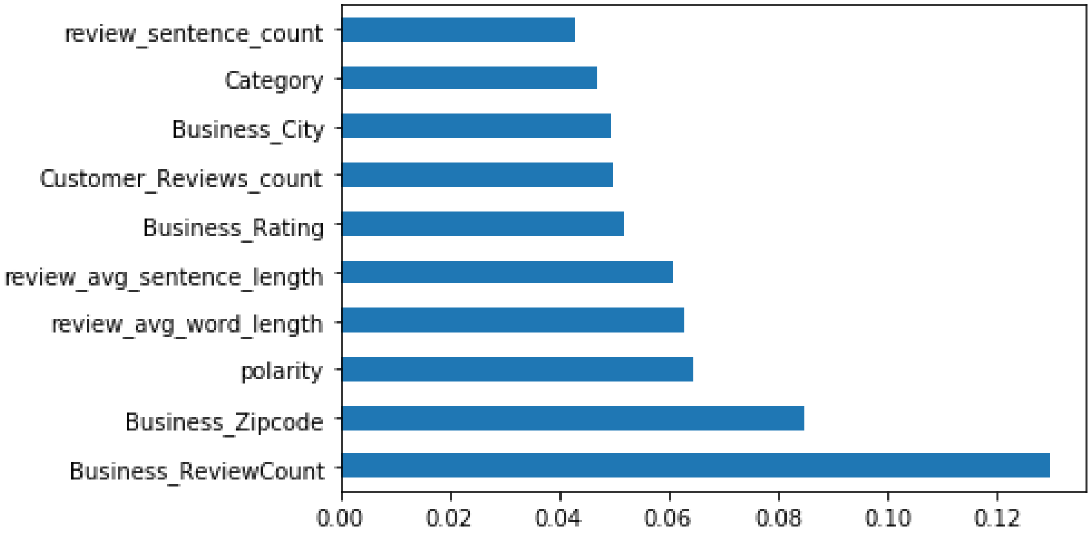


Fig 9 Important features in the model

Figure 9 describes the model's top ten most critical features. Business Review count, Customer Review sentiment, and Business Rating features play a key role in the model, as we previously discussed. We can also see that the zip code of the business location is an important feature in the classification model.

4.5 Oversampling with SMOTE :

SMOTE (synthetic Minority Oversampling TEchnique) consists of synthesizing elements for minority class, based on those already exists. It works randomly by picking a point from the minority class and computing the k-nearest neighbors for that point, the synthetic points are added between the chosen point and its neighbors[7].

In the Fig 10 we can see that synthetic samples are created for the minority class by using the k-nearest neighbors' algorithm.

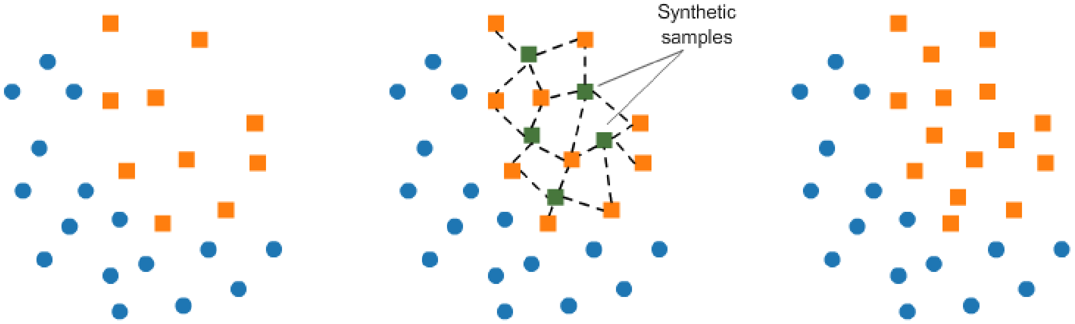


Fig 10: Oversampling with SMOTE

4.4.1 Models performance with oversampling SMOTE data:

The minority class 1 sample will become equivalent to the majority class after oversampling with smote, 79780 samples for each class.

We obtained the following model results by applying ensemble algorithms to the above oversampled dataset.

```
           Model     precision    recall  f1-score   Accuracy
   ---------------------------------------------------------------

    Random Forest          0.93      0.89      0.91    0.91
          XGBoost          0.94      0.86      0.90    0.90
Gradient Boosting          0.88      0.79      0.83    0.84
```

Table 3: Models performance with Oversampling Smote data.

From the Table 3 we can see all model's performance and their results for random forest we got the best accuracy and F1 score as 0.91.

5. Results:

Table 4 shows how the models compare in terms of performance. For Random forest, we got a high Model accuracy of 0.86 and an f1- score of 0.61. We plotted the precision and recall curves to verify the model performance because our data set was imbalanced, and we got a high AU PRC value of 0.63 for the random forest model.

Model	Precision	Recall	F1-Score	Accuracy	AU PRC
Random Forest	0.70	0.43	0.61	0.86	0.63
Gradient Boosting	0.63	0.41	0.57	0.85	0.57
XGBoost	0.53	0.63	0.58	0.83	0.59

Table 4: Models Performance comparison 1

We used the SMOTE oversampling method, in which we created synthetic data points to balance the data set. After oversampling, the dataset became balanced, and we were able to obtain the following results using the models as shown in Table 5. All the models performed better with a balanced dataset. We plot the ROC curve to verify the model performance of a balanced dataset, and we got the model accuracy as 0.91 and AU ROC value as 0.96 for the random forest.

Model	Precision	Recall	F1-Score	Accuracy	AU ROC
RandomForest_SMOTE	0.97	0.84	0.90	0.91	0.96
Gradient_Boosting_SMOTE	0.98	0.78	0.87	0.88	0.94
XGBoost_SMOTE	0.96	0.85	0.90	0.91	0.94

Table 5: Models Performance comparison 2

6. Deliverables:

This project's deliverable is a binary classification model that can classify the types of Business Reviews to which the business owner is likely to respond. After training the model with review data, we can provide new review data and determine whether the business owner will respond to the review. This model can be used by company owners or consumers who want to engage in future interactions.

7. Self- Assessment:

When we first began this project, I didn't have a complete understanding of the following points.

- Techniques for dealing with dynamic web applications to collect data via web scraping.
- How to Deal with an imbalanced Dataset
- Using ensemble algorithms and tuning the model to achieve the best performance.
- Choosing the appropriate metrics to assess the model's performance.
- Story-telling abilities.

Now I know how to deal with the challenges listed above. I have put the skills I learned in DSA courses like Computing Structures, Intelligent Data Analytics, Statistics, Machine Learning Practice, and Text Analytics to use. Apart from that, I have self-taught myself Web Scraping techniques using various tools such as Selenium and working with ensemble Algorithms and imbalance datasets.

For this Project I worked with Dr. Naveen Kumar, Assistant Professor of Management Information Systems, University of Oklahoma.

8. References:

1. Yelp; About Yelp https://blog.yelp.com/2018/01/yelp_category_list#:~:text=There%20are%20more%20than%201%2C200,the%20list%20keeps%20on%20growing.
2. Ryan Erskine ; 20 Online Reputation Statistics That Every Business Owner Needs To Know ; https://www.forbes.com/sites/ryanerskine/2017/09/19/20-online-reputation-statistics-that-every-business-owner-needs-to-know/?sh=ace73c4cc5c9
3. Gender-guesser ; https://pypi.org/project/gender-guesser/
4. Parthvi shah; Sentiment Analysis using TextBlob ; https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524
5. Sharoon Saxena; A Beginner's Guide to Random Forest Hyperparameter Tuning; https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/
6. AARSHAY JAIN; Complete Machine Learning Guide to Parameter Tuning in Gradient Boosting (GBM) in Python; https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/
7. XGBOOST ; https://www.analyseup.com/python-machine-learning/xgboost-parameter-tuning.html
8. SMOTE; Resampling strategies for imbalanced datasets; https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Appendix 1:

S.no	Features	Data Type	Description
1	Business_Name	Text	Name of Business
2	Business_Address	Address	Address of Business
3	Business_ReviewCount	Integer	Total Reviews count of Business
4	Business_Rating	Categorical	Rating of Business
5	Business_Photos_Count	Integer	Count of photos that Business Owner posted
6	Business_Timings	Time	Business operational timings
7	Business_Claim_status	Categorical	Business Claim status
8	Customer_Name	Text	Number of the customer who has given review
9	Customer_Friends_count	Integer	Number of friends that customer has in yelp website
10	Customer_Reviews_count	Integer	Number of reviews that customer posted in yelp website
11	Customer_Photos_count	Integer	Number of Photos that Customer Posted in Yelp website
12	Customer_Elite	Categorical	Customer Elite status in Yelp
13	Customer_Elite_Year	Integer	Customer part of Elite group year
14	Customer_Rating	Categorical	Rating given by customer for business
15	Customer_Review	Text	Review of customer
16	Customer_Review_Date	Date	Date on which the customer given review
17	Customer_Review_Uploaded_Photos	Integer	count of photos that customer uploaded
18	Customer_Review_Useful	Integer	count of people responded for customer review as useful
19	Customer_Review_Funny	Integer	count of people responded for customer review as funny
20	Customer_Review_Cool	Integer	count of people responded for customer review as cool
21	Business_response_By	Text	Business response given by
22	Business_response_Date	Date	Date on which the Business owner responded for review
23	Business_Response_for_Review	Text	Business owner response for customer review.
24	Business_Response	Categorical	Business owner response status.