Project Report
CSCI-P556: Applied Machine Learning Fall 2019

Building a Restaurant Success Model Using Yelp Data

Rushikesh Gawande (rgawande)
Ruta Pravin Utture (rutture)
Sumeet Sarode (ssarode)
Priyanshi Gupta (prigupt)

December 2nd, 2019

1 Introduction

Running a successful business becomes enormously difficult when it comes to maintaining a good reputation as well as tackling competition. For restaurants, ratings on an application like Yelp reveals it's quality and services as well as play an important role in attracting customers. The increasing significance of online reviews also gives rise to fraudulent reviews where competitors may manipulate readers' opinions for their advantage.

Our project focuses on predicting the ratings and evaluating the popularity of a restaurant after detecting fake reviews and disregarding them. Our project will provide a comparative analysis of different supervised learning algorithms for predictions as well as fake review detection by using data from Yelp. We aim to come up with an accurate model for prediction at the end of our endeavors.

2 Data

Yelp is a public company that publishes crowd-sourced reviews about a particular business. The data set used in this project is business directory service and crowd-sourced review forum data. The data is collected from yelp.com[1] which is a collection of multiple data files such as Business Data, User Data, and Review data. The data altogether sums up to approx. 200000 instances with almost 40 features. Business data set contains all the information about a business, most important of which are- the ID of the business, name, and its star rating.

User data contains all the user-relevant information such as the user-ID, number of reviews posted by that user, the votes, and the average rating the user gives to the business. Review Data contains the ID of the user that has posted the review, the business for which the review is cast, the stars given by the user in that review, date of posting the review, and the votes received by it.

3 Related Work

As Yelp is one of the major customer rating directory services in the United States of America, it becomes a go-to platform for a huge customer base. A lot of research has already been conducted on Yelp data set covering areas like fake review detection, the effect of Groupon on Yelp rating prediction [6], research on why people use Yelp, Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews [8], etc. This implies that reviews are an imperative part of Yelp which encourages more exploration to discover new patterns. It is due to this important role which reviews and ratings have, that unfortunately, many businesses have become victim to fraudulent reviews. Previous studies have inferred that Yelp uses a behavioral-based approach for fake reviews filtering which has been proven efficient.[9] As the Yelp method to identify fake reviews is not publicly available many attempts have been made to employ a good algorithm for fraud detection on different datasets that have been bifurcated into 3 categories[10].

1) Deviations among Rating-Based Detection: In this category, the review is used to calculate the rating which intern is compared with actual rating to find deviation. This method has been incorporated by S. P. Algur and J. G. Biradar in [11].
2) Review Content along with User Behavior-Based Detection: This category uses the features of both the review as well as user behavior for fake review detection used by N. Jindal and B. Liu [12]
3) Review Content-Based Detection: This technique is based on detecting a discrepancy in the content of the review.

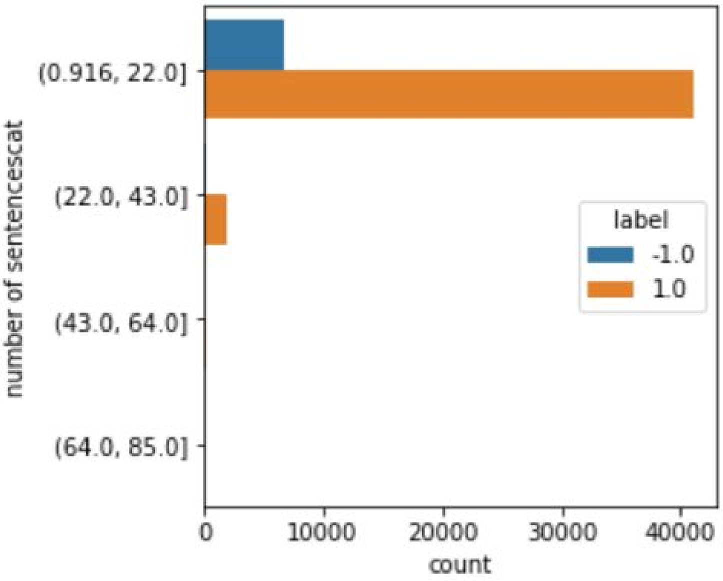
In this project, we will be starting with fake review detection of the Yelp restaurant dataset which will be further used in prediction of rating as well as the popularity of restaurant these procedures to get efficient predictions.

4 EDA

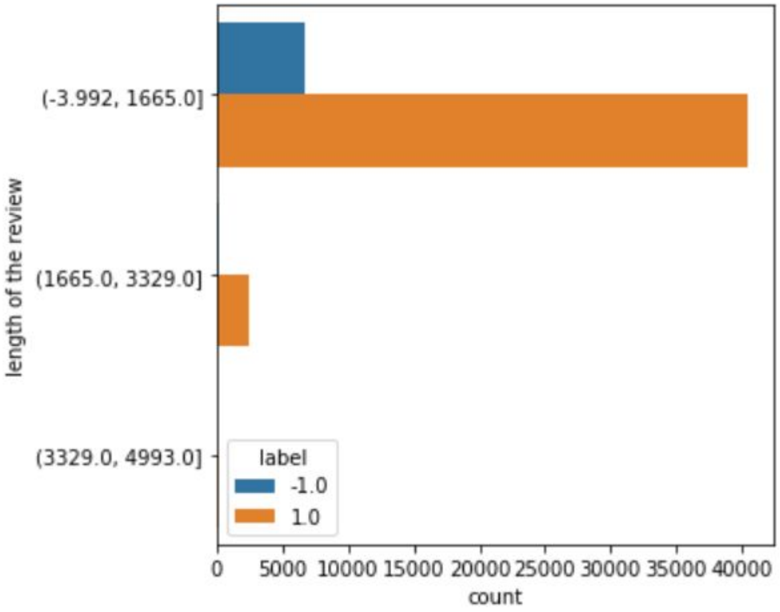
Since we had a huge number of reviews in our dataset, therefore before extracting the unigram and bigram features we decided to analyze the most conspicuous words from the reviews. We used the wordcloud package to extract these words :

We performed further analysis of the data to get some meaningful insights to bifurcate real reviews from fake reviews before training our model on it:

Graph 1: Shows us that reviews with a large number of sentences are more likely to be real reviews.



Graph 2: Shows us that reviews with a long length of review are more likely to be real review.



5 Research questions

Using the Yelp restaurant dataset provided by yelp.com, we want to be able to answer the following questions:

1. How to know if certain reviews are genuine and reliable to the user? Classify whether a given restaurant review is fake or not.
2. How to help the user identify which restaurant is better compared to the others? Predict the overall star rating of the restaurant based on the features of the restaurant.
3. How to help the restaurant owners know how popular their restaurant is? Based on how the restaurant reviews and ratings vary, evaluate the general trend of how the restaurant is performing.

A Research Question One:

Preprocessing: We decided for the first research question, we're going to train our model against a dataset which contains only reviews and it's semantic features along with the label of the review. The reviews are related to restaurant reviews but are not similar to the restaurants we have in testing. We were able to label the data and pre-process the dataset. In the second phase of the project, the main concentration was to generate features for semantic analysis using the reviews of different customers.

We used the reviews to generate the following new features in our dataset:

- **Structural feature extraction:** length of the review, average word length, number of sentences, average sentence length.
- **POS percentages:** All the words in each review are tagged using 45 grammatical tags which explains the use of that word in the review.
- **Semantic features:** Percentage of Positive polarity words, Percentage of Negative polarity words. The polarity of each word in a review is calculated using Natural Language ToolKit.
- **Unigram and Bigram feature extraction:** We have used Term Frequency- Inverse Document Frequency (TF-IDF) to extract most significant unigram and bigram features (100 each) using sklearn.feature_extraction.text package in python. The intuition behind finding the significant words in a review is that the word that appears in almost all reviews is less likely to be significant. The assigned weight to each word in the entire document by IDF shows how rare or common a word actually is.

Feature Engineering: We first started by trying to scale the data by using feature scaling methods of MinMax scaler, Standard scaler and Robust Scaler but did not have much effect on the models. However, we normalized the data using Normalizer from sklearn library which turned out to be little effective. We also tried to reduce the dimensions using PCA, however as we had 135 features for this one, it turned out that dimensionality reduction was not having any significant impact.

Algorithms: Since in this research question we have to classify the reviews as fake or authentic, we tried various classification algorithms to get close to accurate predictions.

1) Naïve Bayes Algorithm: To start with, first we used multinomial naïve bayes classifier which is mostly used for document categorization when the features are related to the frequency of words, length etc. of the document. We implemented it with the help sklearn.naive_bayes module of scikit library.

2)Decision Tree: The Naïve Bayes was just a baseline model and the accuracy was not enough so we tried implementing decision tree algorithm using sklearn.tree module of scikit library. We hoped that decision tree should give a better result than Naive Bayes and it did.

3)Logistic Regression: Logistic regression was implemented using sklearn.linear_model where we gave 1000 value to max iteration parameter and solver 'lbfgs' after comparing the different combinations of iterations and solver.

4)Random Forest: Since we got some better results with decision tree algorithm, we decided to optimize it further. As random forest algorithm outperforms decision tree algorithm, we decided to try random forest algorithm. We implemented this algorithm using sklearn.ensemble module.

5)Neural Networks: At last, we implemented the artificial neural networks on our dataset with the hope of getting the best accuracy. We tried different architectures and finally, we were able to achieve a better accuracy containing three hidden layers with 128 neurons having Relu activation function and 0.2 dropout. The final layer, we applied softmax activation function.

B Research Question Two:

Preprocessing: We worked on the data we got from Kaggle which contains various tables contributing to the attributes of a business. Since the data we got wasn't only catered to restaurants, a lot of the features present were also not related to our research question. The overall process of cleaning the data included segregating businesses that are restaurants from the rest. We merged columns from multiple tables from different datasets. We had to dissolve some categories linked which didn't pertain to a restaurant. Finally, we aggregated all the columns and performed one-hot encoding on the data.
We did principal component analysis for dimensionality reduction by trying different sets of combinations of features and figure out what gives us optimum results. We also tried scaling the features but it was not making a significant impact on the results.

Algorithms :
So in this research question we had to predict the star ratings for the restaurant so we tried various regression algorithms to get the results.

1)Multiple Linear regression: To start with, first we used multiple linear regression which is mostly used as a baseline model for regression problems. We implemented it with the help sklearn.LinearRegression module of scikit library.

2)Decision Tree Regressor: The multiple linear regression was just a baseline model and the accuracy was not enough so we tried implementing decision tree algorithm using sklearn.tree module of scikit library. We hoped that decision tree should give a better result than linear regression but that was not the case. The decision tree was performing poorly as compared to our linear regression.

3)Random Forest Regressor: Since we were not getting satisfactory results with the decision tree algorithm, we decided to optimize it further. As random forest algorithm performs ensemble bagging technique and reduces variance of individual decision trees, we decided to try implementing it. We implemented this algorithm using sklearn.ensemble module and it gave some better results.

4)Neural Networks: We implemented artificial neural networks on our dataset using keras library. We tried different architectures and we decided on the architecture which had three hidden layers with Relu activation function and 128 neurons each. The output layer had linear activation function.

5)K-nearest neighbors: K-nearest neighbors was implemented using sklearn.neighbors module where we gave 5 value to number of neighbors after comparing the results we got for different values of neighbours. It gave much better results than decision tree and random forest.

C Research Question Three:

Preprocessing: Our dataset consists of reviews of all the businesses, the date on which it was published, which business it belongs to and the label. This process took the most time since there were over 5 million reviews and we had to discard reviews not belonging to a Restaurant Business. At the end we were left with a little over 1.5 million restaurant reviews. We then took the help of the model we built for our first research question to label the review as fake or not. We had to maintain the same features of our first dataset so we also had to generate all the features from the reviews to be able to predict the label of each of the reviews.

We then found the polarity of each review. We used Natural Language Processing for analyzing the sentiment of all these reviews. First, the entire review was broken down into tokens and the positive sentiment and the negative sentiment bearing words were segregated. The polarity of the review is then calculated by calculating the polarities of both these group of words.The compound score i.e the polarity of the review is calculated by combining the combined polarities of the positive and the negative sentiment bearing groups. Combining the sentiment and the polarities of these reviews posted in a specific window of time, along with the number of reviews gives us the trend of the restaurant.

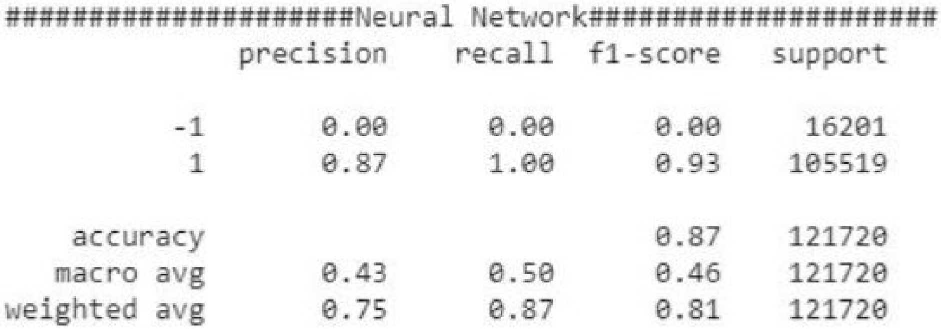
We have built a dashboard in which the user can enter the name of the restaurant and the time period through which they want to calculate the trend for e.g a year from 2006-2017, quarters of a year, monthly trend change. The default year is given as 2016. According to the input obtained from the user, it plots a graph that shows the trend of that restaurant for that time period.

6 Results:

Research Question 1:

Algorithm	Accuracy		F1 Score		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test
Multinomial Naive Bayes	0.59	0.59	0.72	0.72	0.9	0.9	0.6	0.6
Decision Tree	1	0.78	1	0.87	1	0.88	1	0.86
Logistic Regression	0.86	0.8681	0.93	0.93	0.87	0.87	1	1
Random Forest	0.86	0.8682	0.93	0.93	0.87	0.87	1	1
Neural Network	**0.86**	**0.8685**	**0.93**	**0.93**	**0.87**	**0.87**	**1**	**1**

Below is the evaluation metrics for the best performing algorithm “Neural Networks”.

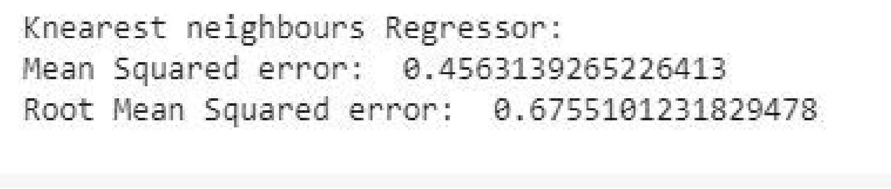


```
#####################Neural Network#####################
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00     16201
           1       0.87      1.00      0.93    105519

    accuracy                           0.87    121720
   macro avg       0.43      0.50      0.46    121720
weighted avg       0.75      0.87      0.81    121720
```

Research Question 2:

Algorithm	MSE		RMSE	
	Train	Test	Train	Test
Multiple linear regression	0.56	0.57	0.74	0.75
Decision Tree Regressor	0.035	1.093	0.18	1.045
Random Forest Regressor	0.114	0.62	0.338	0.78
K-nearest Neighbours	0.443	0.455	0.66	0.67
Neural Network	**0.522**	**0.564**	**0.72**	**0.75**

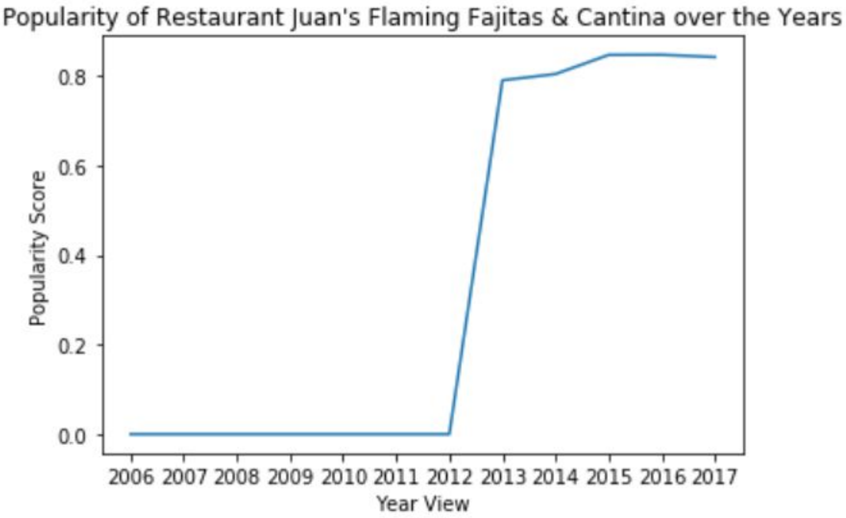
Below is the MSE and RMSE for the best performing algorithm “K-nearest Neighbors”

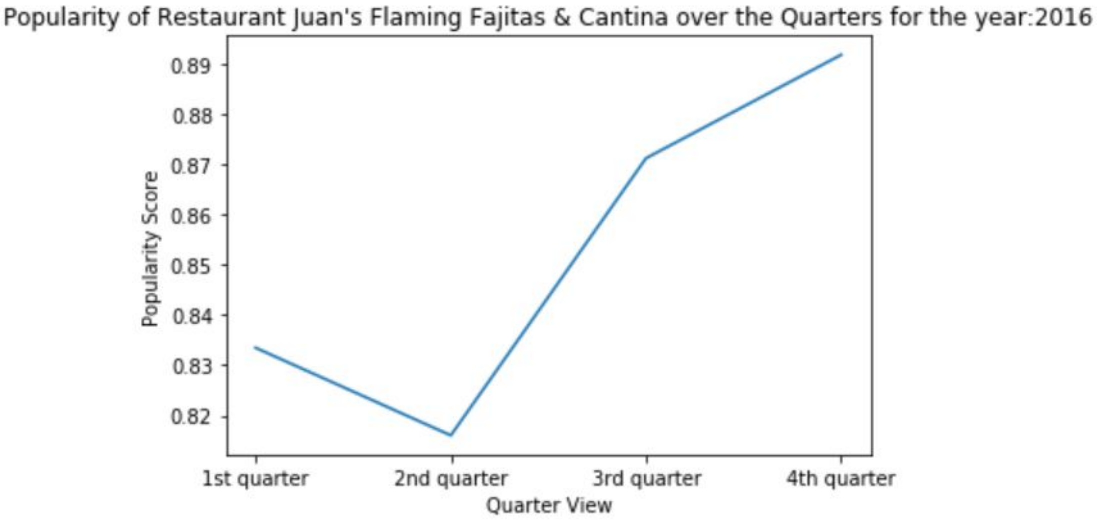


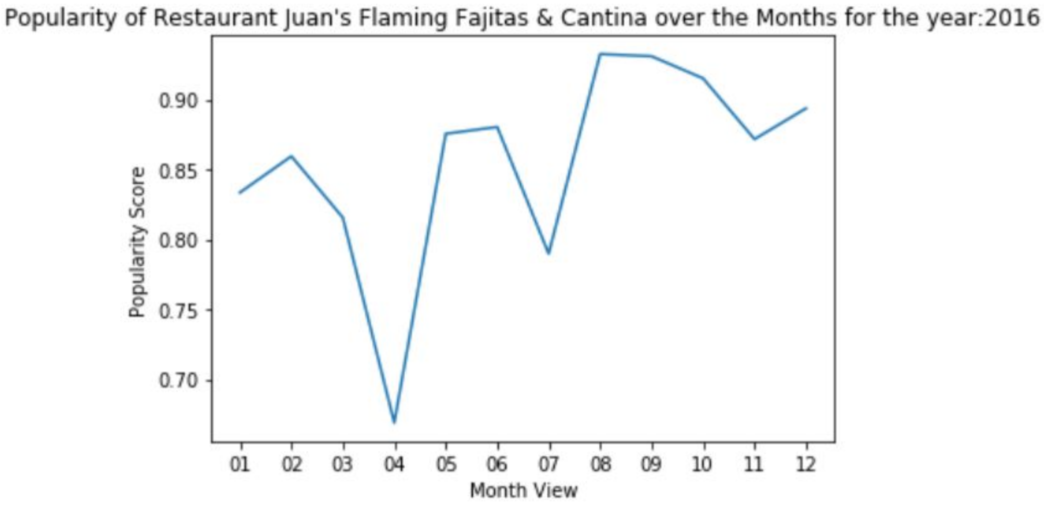
```
Knearest neighbours Regressor:
Mean Squared error:  0.4563139265226413
Root Mean Squared error:  0.6755101231829478
```

.

Research Question 3:







7 Discussion

For classification of the reviews, the outputs and the detailed accuracy reports for Multinomial Naive Bayes, Logistic Regression, Random forest, Decision Tree and Neural Network were studied. Out of all the above mentioned algorithms, Neural Networks gave the highest accuracy. This is because of the ability of neural networks to try and mimic the human brain to produce the classification rules with the help of numerous complex computations it does on the data. Further, we tried a lot of feature engineering techniques to get the best results and found out that Neural Network worked best with normalization of data. We explored introducing dropout in Neural Networks which gave us a slight spike in the accuracy. For future scope, we could add reviewer centric features to our dataset and leverage them to identify fake reviews.

To predict the star ratings of restaurants, we studied the outputs and mean squared error of training and testing for Multiple Linear Regression, Decision tree, Random Forest, Neural Networks and K-nearest neighbors. The k-nearest neighbor gave us the best results out of all the algorithms with the least mean square error. We tried lots of combinations for the PCA, using different no of features at a time, and chose the best outcome out of that. Our decision tree was overfitting a lot and performing worse on testing data, so we tried to regularize it but had no luck. We implemented dropout in the Neural Network to avoid the same overfitting problem, which performed quite well.

We decided to classify the reviews into fake or genuine using Logistic Regression model for the third research question. Even though Neural Networks was showing a slightly higher accuracy we made this decision because we had to label a large quantity of reviews and we wanted to achieve this quickly. Logistic Regression was able to do this in significantly lesser time for a large volume of reviews and it still maintained a good accuracy with it and thus we used this for creating our dataset for Evaluating the trend of restaurants. For future scope, we could create a dashboard for the Restaurant user in which they can toggle between different granularity of views.

8 Conclusion

We were able to answer our first research question by classifying the reviews as fake or not which is most crucial when it comes to a business directory service like yelp which is highly based on user reviews. We were able to answer our second research question by predicting the star rating of the restaurants based on all the features of the restaurant which proves very beneficial to the users availing the service. We were able to answer the third question by predicting the trends in the restaurant based on how customers reviews are varying in a given span of time which proves to be beneficial for the restaurant owners.

References:

1. https://www.yelp.com/dataset/documentation/main
2. https://www.kaggle.com/yelp-dataset/yelp-dataset/version/6
3. http://odds.cs.stonybrook.edu/yelpzip-dataset/
4. http://cs229.stanford.edu/proj2017/final-reports/5244334.pdf
5. http://cs229.stanford.edu/proj2017/final-reports/5229663.pdf
6. https://arxiv.org/pdf/1202.2369.pdf
7. https://arxiv.org/pdf/1605.05362.pdf
8. file:///F:/AML%20Project/1709.08698.pdf
9. What yelp fake review filter might be doing?. In ICWSM. 6006-30389-1-PB (1)
10. https://pdfs.semanticscholar.org/6005/58b138f32b211ffc71bf918c2005a9e4e0e5.pdf
11. https://ieeexplore.ieee.org/document/7456871
12. https://ieeexplore.ieee.org/abstract/document/4470288/references#references