CS410 Fall 2023 - Project Final Report

Project: Yelp Customer Review Sentiment Analysis

Team Name: CaffeineCrew

Team members

- Goutam Debnath (Captain) - goutamd2@illinois.edu
- Naveen Baskaran - nc42@illinois.edu
- Rohit Narula - rnarula2@illinois.edu
- Umesh Kumar - umesh2@illinois.edu

Abstract

Sentiment analysis (also called opinion mining) is a natural language processing (NLP) technique used to determine whether the data is positive, negative, or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback and understand customer needs. Sentiment analysis focuses on the polarity of a text (positive, negative, neutral) but it also goes beyond polarity to detect specific mood (angry, happy, sad), urgency (urgent, not urgent) or even intentions (interested, not interested). Depending on how we want to interpret customer feedback and queries, we can define and tailor the categories to meet our sentiment analysis needs. These sentiments become extremely gainful for organizations, businesses, states, and people. In this project, we are going to attempt to focus on the polarity of a customer review. Our goal is to build a sentiment analysis model that predicts whether a user liked a business or not, based on their reviews on Yelp.

Data Selection And Cleaning

We utilized the open-source user review dataset from the Yelp Dataset. We leveraged the open-source public dataset found on the Yelp website which has customer commented reviews for various products and services offered by the businesses like car rentals, hotels, restaurants, attractions, coffee shops, fast foods to name a few. Yelp provides access to a large collection of public datasets through their Yelp Open Dataset program. We downloaded the JSON format of the data from their site.

The main file that was of interest for our analysis is the **yelp_academic_dataset_review.json** file. The primary task before performing any model building is to perform Data cleaning and do Exploratory Data Analysis. This includes various tasks from filtering the sample size of the dataset, interpolating null values (if any), transforming data to the use case of our need. On the

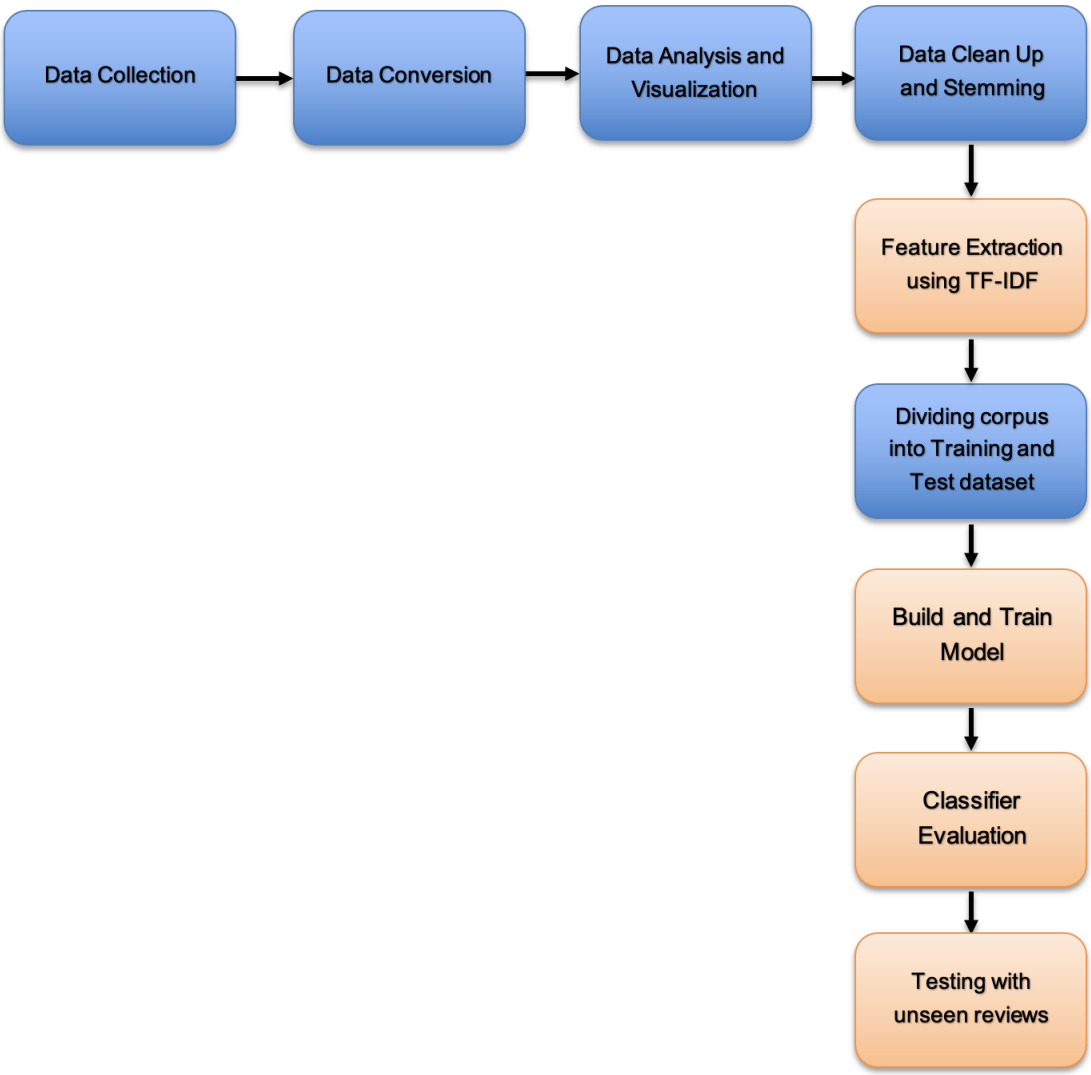
initial review of the downloaded dataset, we found the below. The JSON file data was too large, and we converted the JSON file to a csv file using python pandas data frame. Once the csv file was available, we loaded it to the Open Refine tool which is capable of processing large volumes of files along with data transformations, format conversions and text clustering capabilities. The dataset had around 200,000 rows with various columns like business id, review id, reviewer id, timestamp of the review, star rating on a scale of 1 through 5 and user comments on the reviews (like cool, funny, useful).

We filtered the dataset to select 70,000 rows for this analysis task. We checked this dataset for any missing values. After exploration, the dataset was found to be cleaner and there were no missing values. All columns on all rows of the dataset have been populated. There were 9 features with string and number types.

Implementation

Workflow

Below diagram shows various tasks involved in sequence to complete this project.



Data Exploration

The cleaned Yelp dataset contains 10,000 reviews with features review_id which is unique id for the posted review, user_id which is the unique id of the user writing review, business_id which is the unique id of the business being reviewed, stars which carrying values from 1 through 5 as rated by the customers for the businesses, comment reaction given by other users which are cool, useful and funny, text which is the review comment that can be unstructured and date which is the date the review was posted to the Yelp website.

We created a bar plot to get a sense of how the star ratings were distributed between 1 to 5. The plot shows that 4 and 5 stars reviews are higher in number compared to the remaining of the ratings.

```
Out[8]: <Axes: xlabel='stars', ylabel='count'>
```

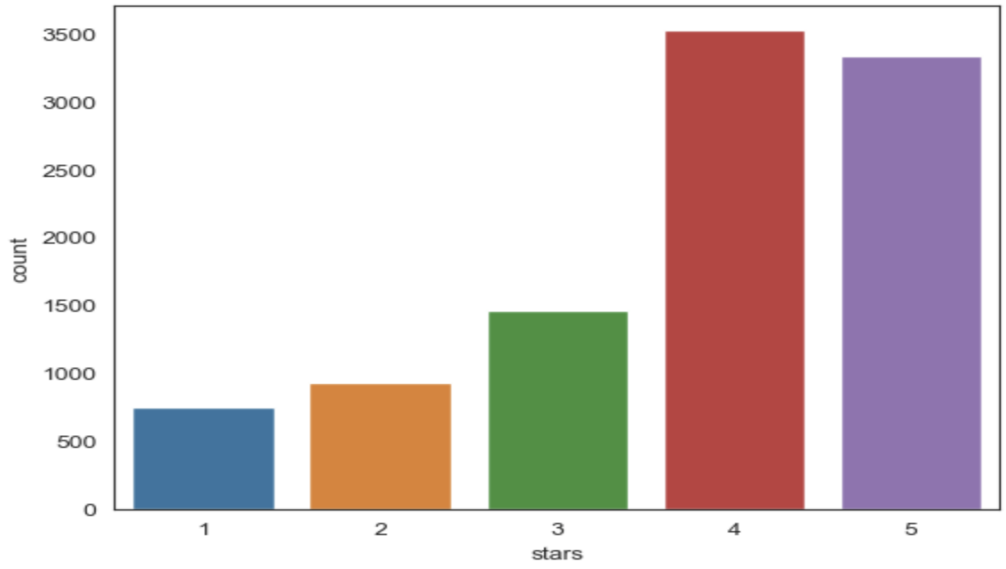


Figure 2: Bar chart between review count and star ratings (1 - 5)

Data Analysis Through Visualization

To understand the relationship between the length of the text review and stars ratings received, we added a new column “text_length" to the dataset as shown in the Figure 3.

	review_id	user_id	business_id	stars	useful	funny	cool	text	date	text_length
0	KU_O5udG6zpxOg-VcAEodg	mh_-eMZ6K5RLWhZyISBhwA	XQfwVwDr-v0ZS3_CbbE5Xw	3	0	0	0	If you decide to eat here, just be aware it is...	7/7/18 22:09	513
1	BiTunyQ73aT9WBnpR9DZGw	OyoGAe7OKpv6SyGZT5g77Q	7ATYjTIgM3jUlt4UM3IypQ	5	1	0	1	I've taken a lot of spin classes over the year...	1/3/12 15:28	829
2	saUsX_uimxRlCVr67Z4Jig	8g_iMtfSiwikVnbP2etR0A	YjUWPpI6HXG530lwP-fb2A	3	0	0	0	Family diner. Had the buffet. Eclectic assortm...	2/5/14 20:30	339
3	AqPFMleE6RsU23_auESxiA	_7bHUi9Uuf5__HHc_Q8guQ	kxX2SOes4o-D3ZQBkiMRfA	5	1	0	1	Wow! Yummy, different, delicious. Our favo...	1/4/15 0:01	243
4	Sx8TMOWLNuJBWer-0pcmoA	bcjbaE6dDog4jkNY91ncLQ	e4Vwtrqf-wpJfwesgvdgxQ	4	1	0	1	Cute interior and owner (?) gave us tour of up...	1/14/17 20:54	534

Figure 3: Modified sample review dataset

We plotted between review text length and stars received to the review using the Python Seaborn library. It shows that the length of the reviews are higher towards 4 and 5 stars. However, distribution of the text length is similar across the rest of the ratings.

```
<seaborn.axisgrid.FacetGrid at 0x126ab7190>
```

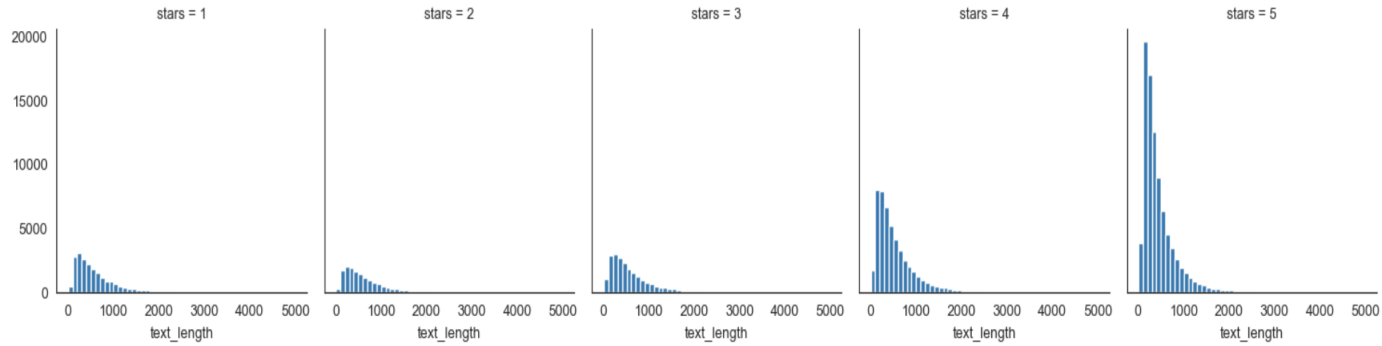


Figure 4: Plot between review length and stars

We then created a box plot between review text length and stars received to the review using python seaborn library. It shows 1 and 2 stars reviews have much longer text however there are many outliers, which can be seen as points above the boxes.

```
Out[9]: <Axes: xlabel='stars', ylabel='text_length'>
```

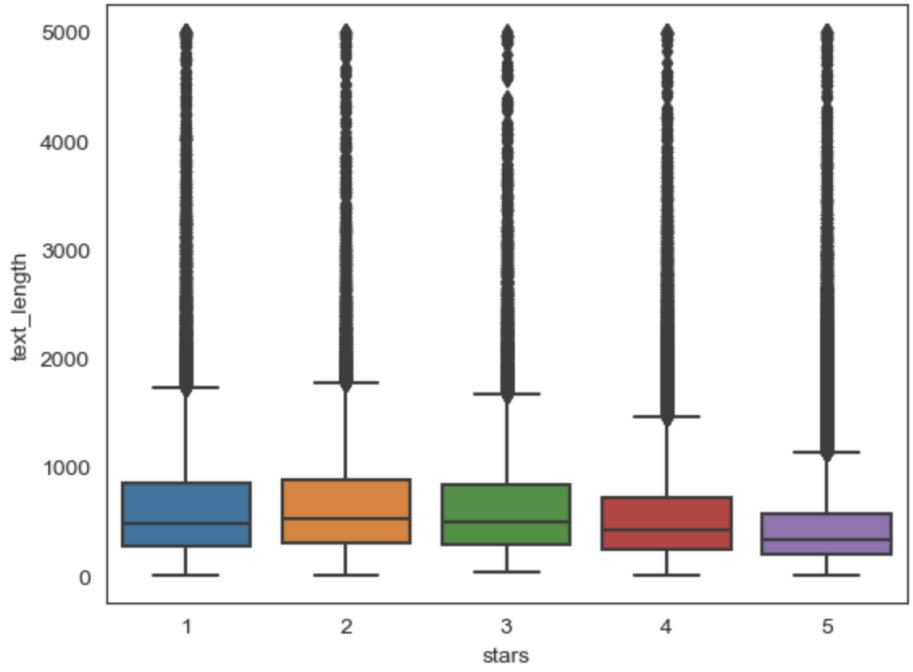


Figure 5: Box plot to correlate stars to text length of reviews

Further we grouped the data by review stars to know the correlation between text length and review features which are recorded as cool, useful and funny. We used python pandas library to calculate the correlation and then created a heatmap using seaborn library. The plot shows funny reviews are strongly correlated with the useful reviews.

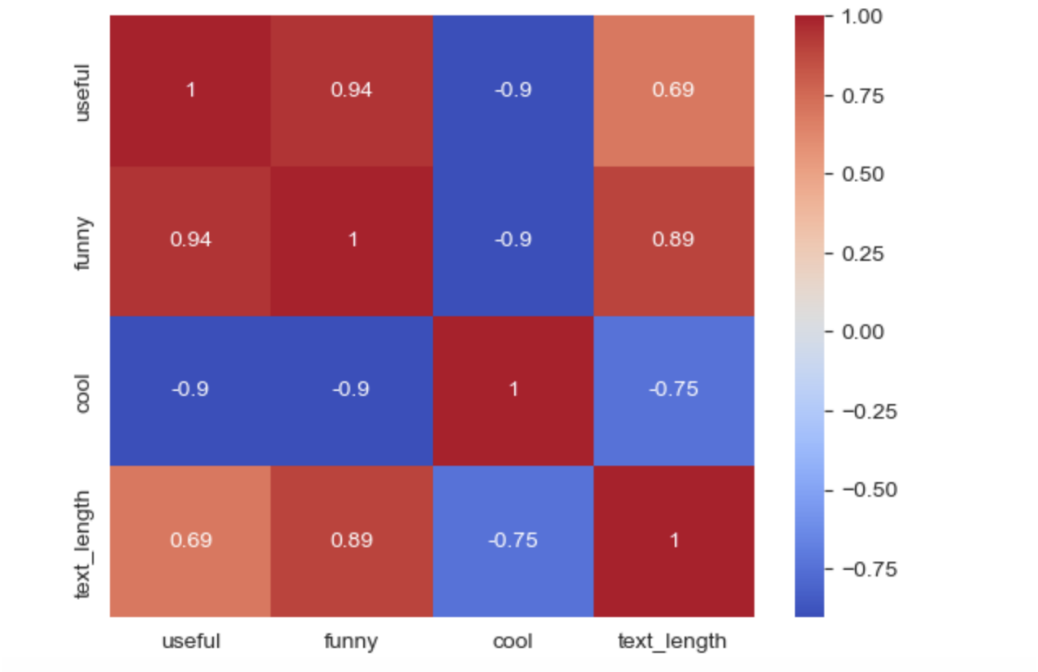


Figure 7: HeatMap to correlate review with features (cool, useful or funny)

Data Preprocessing

Data preprocessing is a crucial step which involves cleaning, transforming, and preparing raw textual data for analysis. We used a range of NLP techniques (as mentioned below) to prepare data that is suitable for the NLP algorithm to extract meaningful information.

Text Cleaning: We cleaned the data by removing irrelevant information, such as special characters and punctuation. We used regex match to perform this task.

Lower Casing: We converted the text to lowercase to ensure consistency.

Tokenization: We performed the tokenization of the review. It is the process of breaking down text into individual units or words. We used the NLTK (Natural Language Toolkit) library in python for performing tokenization of the text review.

Stop Words Removal: Stop word removal is an important text preprocessing step that involves identifying and removing common or non-content words from the text data. These words (such as “the,” “and,”, “is”, "a" “an”) are often considered to have little or no meaning and contribute to noise in the data.

Stemming: We have used the stemming algorithm PorterStemmer from NLTK library to reduce the words to their root or base form. For example “computing” becomes “compute”, “running” becomes “run”.

The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Stemming helps reduce redundancy as most of the time the word stem and their inflected/derived words mean the same. The idea of stemming is a sort of normalizing method. Many variations of words carry the same meaning, other than when tense is involved. The reason why we stem is to shorten the lookup, and normalize sentences.

```
def preprocess_text(text):
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = text.lower()
    tokens = nltk.word_tokenize(text)
    tokens = [word for word in tokens if word not in set(stopwords.words('english'))]

    # Apply Porter Stemmer
    stemmer = PorterStemmer()
    stemmed_tokens = [stemmer.stem(word) for word in tokens]

    preprocessed_text = ' '.join(stemmed_tokens)
    return preprocessed_text
```

Figure 8: Code snippet for preprocessing of data

Feature Extraction

Feature extraction is an essential step in natural language processing (NLP) tasks, as it transforms raw textual data into a numerical representation that can be further processed by a classifier effectively. We used one of the most powerful feature extraction techniques, Term Frequency-Inverse Document Frequency (TF-IDF). We have utilized TF-IDF by importing TfidfVectorizer from scikit-learn library to vectorize the review text.

Model building for Classification and Evaluation

Building and Training Model

We wanted to create a predictive mode model that can categorize the data points into predefined classes. For simplicity, we filtered the dataset to select the first 3000 reviews that had ratings of 1 or 5. Then we used 80% as training data and 20% as test data.We used the “text” feature for training and the “stars” feature as a label from our training dataset.

Next, we built a classifier model using four below machine learning algorithms and trained the model with the training dataset. Using this trained model, we were able to predict the star ratings of 1 or 5 for the review from the test dataset.

1. **A Support Vector Classifier:** SVC is a type of supervised machine learning model that belongs to the family of Support Vector Machines (SVMs).For this project we considered only two classes (positive rating of 5, negative rating of 1) and hence this becomes a binary classification problem here. We defined SVC with a linear kernel and default regularization parameter of 1. Once the model is trained, we use this fitted model to predict the classes using the test dataset and evaluate the model by calculating the model score.

2. **Multinomial Naïve Bayes:** Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article.

3. **Logistic Regression:** Logistic Regression is a statistical method used for binary classification problems, where the response variable is categorical and has two classes (e.g., Positive and Negative). It's a widely used algorithm in machine learning for predicting the probability of an observation belonging to a particular class. This model is trained using the maximum likelihood estimation, which aims to maximize the likelihood of the observed data given the model parameters.

4. **Random Forest:** Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

Model Evaluation:

We use Precision and Recall as the evaluation metric to measure our rating prediction performance. We compare our prediction with the star ratings to determine the correctness of our prediction.

In order to evaluate the model's performance, we utilized the below four metrics.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 Score = 2 * (precision * recall) / (precision + recall)

Precision measures the relevance of the results. Recall is a valuable metric for evaluating NLP classifiers, as it provides an indication of how well the model can identify all positive instances. F1-score is a harmonic mean of precision and recall.

Here, TP (true positives) represents the number of instances that are correctly classified as positive.

TN (true negatives) represents the number of instances that are correctly classified as negative.

FP (false positives) represents the number of instances that are incorrectly classified as positive.

FN (false negatives) represents the number of instances that are incorrectly classified as negative.

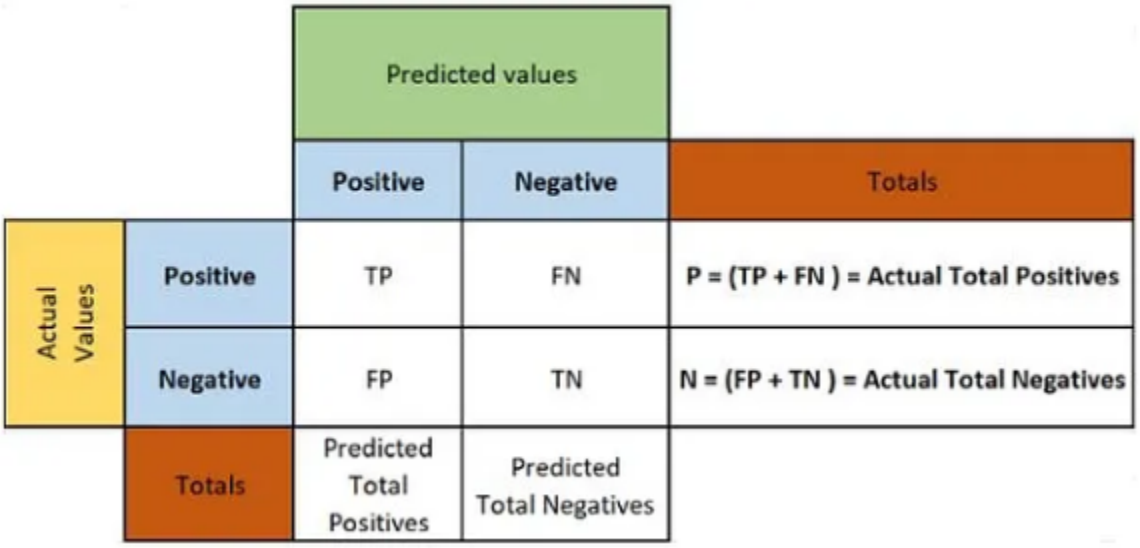


Figure 9: Confusion matrix

We evaluated all four classifier's performance using an earlier created test dataset. We calculated Accuracy, Precision, Recall and F1 Score for every classifier as shown below.

```
Number of total samples: 3000

Support Vector Classifier Classifier (Based on Reviews):
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
Classification Report:
               precision    recall  f1-score   support

           1       0.94      0.78      0.85       104
           5       0.96      0.99      0.97       496

    accuracy                           0.95       600
   macro avg       0.95      0.88      0.91       600
weighted avg       0.95      0.95      0.95       600
```

Figure 10: Support Vector Classifier: Performance Metrics

```
Multinomial Naive Bayes Classifier (Based on Reviews):
Accuracy: 0.86
Precision: 0.87
Recall: 0.86
Classification Report:
              precision    recall  f1-score   support

           1       0.95      0.18      0.31       104
           5       0.85      1.00      0.92       496

    accuracy                           0.86       600
   macro avg       0.90      0.59      0.61       600
weighted avg       0.87      0.86      0.81       600
```

Figure 11: Multinomial Naive Bayes Classifier: Performance Metrics

```
Logistic Regression Classifier (Based on Reviews):
Accuracy: 0.92
Precision: 0.92
Recall: 0.92
Classification Report:
              precision    recall  f1-score   support

           1       0.95      0.57      0.71       104
           5       0.92      0.99      0.95       496

    accuracy                           0.92       600
   macro avg       0.93      0.78      0.83       600
weighted avg       0.92      0.92      0.91       600
```

Figure 12: Logistic Regression Classifier: Performance Metrics

```
Random Forest Classifier (Based on Reviews):
Accuracy: 0.91
Precision: 0.92
Recall: 0.91
Classification Report:
              precision    recall  f1-score   support

           1       0.96      0.51      0.67       104
           5       0.91      1.00      0.95       496

    accuracy                           0.91       600
   macro avg       0.94      0.75      0.81       600
weighted avg       0.92      0.91      0.90       600
```

Figure 13: Random Forest Classifier: Performance Metrics

We found that Support Vector Classification (SVC) has a maximum accuracy of 95% and Multinomial Naïve Bayes Classifier has minimum 86% accuracy.

Classifier Evaluation with Visualization

We plotted a Bar chart to compare Accuracy, Precision and Recall of every classifier.

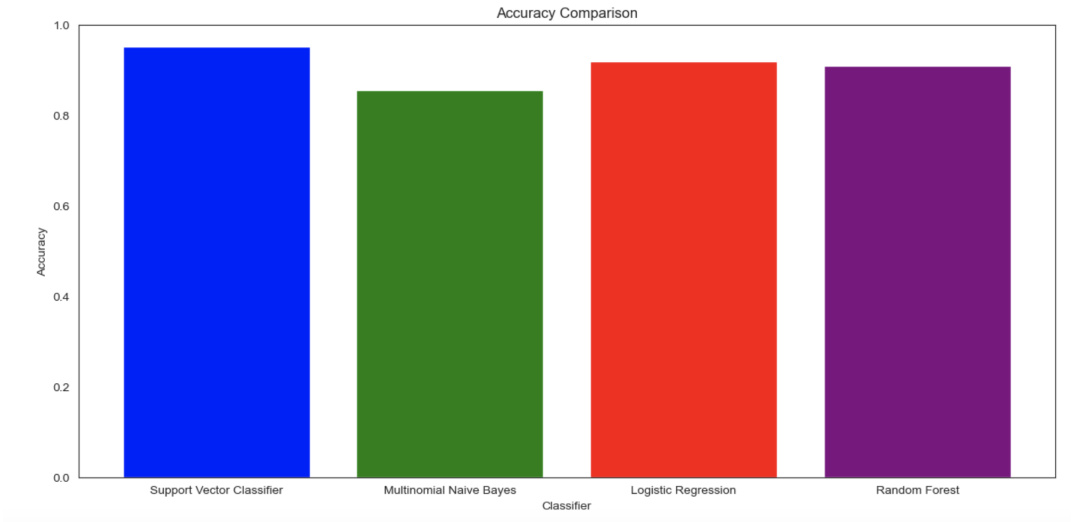


Figure 13: Accuracy comparison

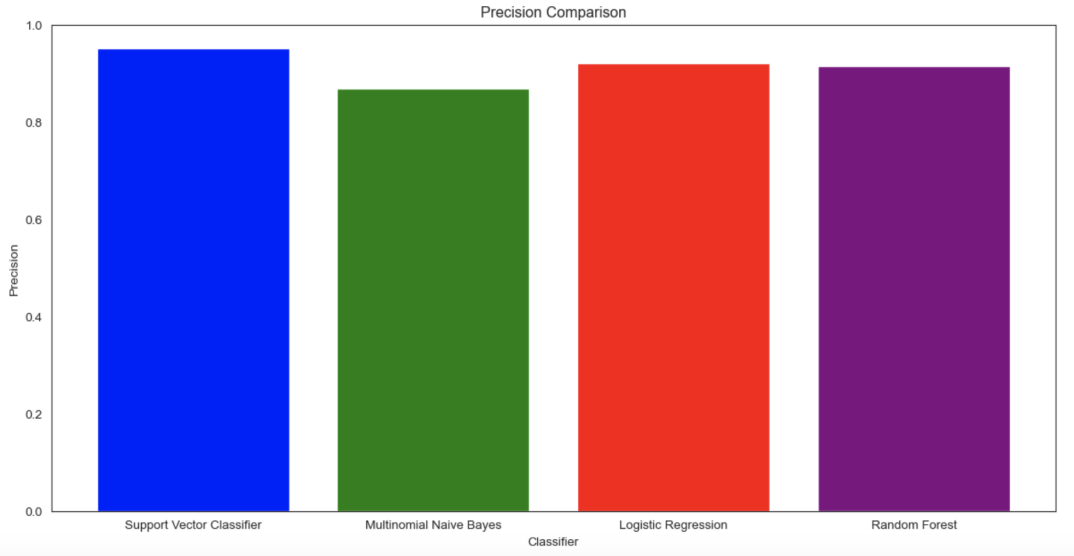


Figure 14: Precision Comparison

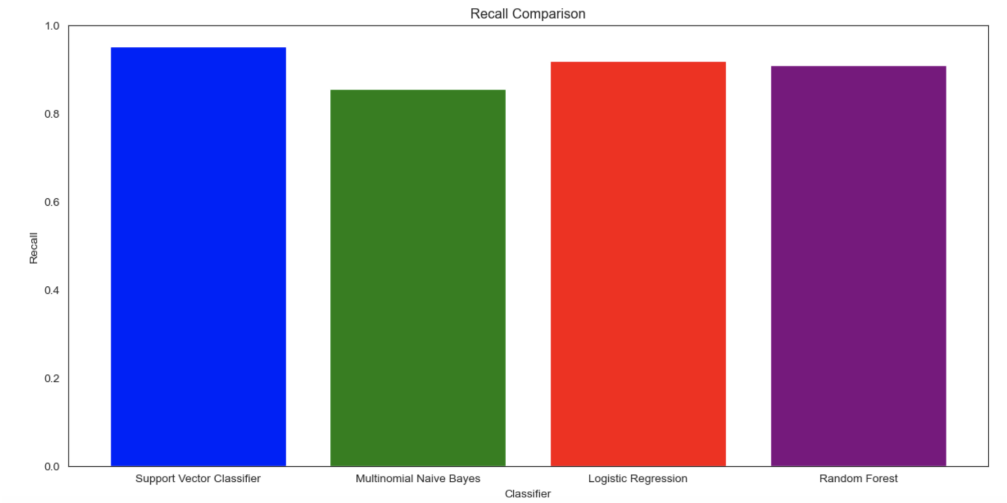


Figure 15: Recall Comparison

Prediction with Unseen Review

We used below two new unseen Yelp reviews to predict the star ratings using every classifier to understand its behavior.

Review #1*: Horrible, they have a scary mascot that scared my grandson and I asked him twice and my waitress for him not to come to our table. Did he listen NO,kept coming back! Also served a drink with a big crack! We will never return*

Support Vector Classifier Predictions:
Predicted Star Rating: 1
Actual Star Rating: 1

Multinomial Naive Bayes Classifier Predictions:
Predicted Star Rating: 5
Actual Star Rating: 1

Logistic Regression Classifier Predictions:
Predicted Star Rating: 1
Actual Star Rating: 1

Random Forest Classifier Predictions:
Predicted Star Rating: 1
Actual Star Rating: 1

Review #2*: Update after initial review: ordered again last night using the yelp eat24 app. Food was ready within 10 minutes of placing the order. I opted for pickup since I was on my way home. The app did not give them my correct phone # and speaking to the great staff at the reception desk they told me sometimes yelp gets the addresses wrong for delivery so my advice is to type your phone # and address (if using delivery) into the comments box just so you get your food as fast as they intend to get it to you.*

This is the best place for sushi in the area. We have frequented other sushi places and this one by far stands above the rest. Was happy to see Saturday night was very busy for them, and my takeout was still ready in under 10 minutes!!

Support Vector Classifier Predictions:
Predicted Star Rating: 5
Actual Star Rating: 5

Multinomial Naive Bayes Classifier Predictions:
Predicted Star Rating: 5
Actual Star Rating: 5

Logistic Regression Classifier Predictions:
Predicted Star Rating: 5
Actual Star Rating: 5

Random Forest Classifier Predictions:
Predicted Star Rating: 5
Actual Star Rating: 5

Conclusion

With the prediction model, we were able to predict star ratings of the unseen review with accuracy up to 90%. Any business trying to understand the customer sentiments based on the star ratings for the given reviews, they will be able to use this tool with higher accuracy. It can help the business focus on the reasons causing the negative reviews and focus on the priorities on getting those negative experiences limited. Additionally they can further focus on engaging the customers with targeted promotional campaigns who give positive reviews.

Potential for Future Work

For SVM, Adjustments and fine-tuning can be done based on the specific characteristics of your data.

We have experimented with various feature selection and supervised learning algorithms to predict star ratings of the Yelp dataset using review text alone. We evaluate the effectiveness of different algorithms based on precision and recall measures. Possible improvement could be extracting additional information from the dataset such as Business Categories and use customized feature sets for each Category, because different word features might be more or less relevant in different Business Categories. Runtime of the algorithm could possibly be

improved by training and testing within each business category, because of a smaller feature set. We could also try using parts-of-speech in the feature selection process to differentiate between the same word features that are used as different parts-of-speech.

References

[1] Rayudu yarlagadda, Sentiment analysis Blog. urytrayudu1.medium.com/sentiment-analysis-for-yelp-review-classification-54b65c09ff7b, 2018

[2] Learn to Code for NLP – Youtube, www.youtube.com/watch?v=edFX28Z7jlM

[3] Yelp Open Dataset: https://www.yelp.com/dataset

[4] Open Refine: https://github.com/OpenRefine/OpenRefine

[5] SVM: https://www.geeksforgeeks.org/support-vector-machine-algorithm/

[6] Logistics Regression: https://www.geeksforgeeks.org/understanding-logistic-regression/

[7] Naive Bayes: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[8] Random Forest: https://en.wikipedia.org/wiki/Random_forest