Sentiment Analysis on Yelp Data

York (Yikai) Fang
Yeran Liu
Sonal Mendiratta
Kevin Seng
Adwait Toro

Table of Contents

1. Background

i. About Yelp

Founded in San Francisco in 2004 by two former Paypal employees, Yelp is an online community platform where users can share reviews of local businesses. It also provides local businesses with opportunities to engage more with their customers. Since its inception, Yelp has seen constant growth (Exhibit 1) and has remained a leading company in a competitive crowd-sourced review market due to its many product innovations. Over the years, they have embedded a reservation and waitlist system and strategically partnered with a food delivery service. Like many other social media networks, Yelp makes almost all of its revenue through targeted advertisements. This has posed questions from many investment analysts on whether this is a sustainable long-term revenue model. In response to this, they have been in the discussion of acquiring Groupon, a deal base site, to potentially add deals to their revenue stream.

ii. About the Dataset

We retrieved the Yelp Dataset directly from them, as they have made it available on open-source for competition and research purposes. For the basis of our analysis, we solely focused on the review and business datasets. The review dataset has close to 7 million reviews while the business dataset has over 190K businesses, with restaurants having the highest representation.

iii. Objective

It is proven by Harvard business school that Yelp review has direct impact on business revenue. Customers are utilizing Yelp as reliable measures of a restaurant's quality. Our objective is to help Yelp or businesses by managing customer relationship through sentiment analysis.

2. Exploratory Data Analysis

Before doing Sentiment Analysis we analyzed the data to understand the user behaviour as well as overall Yelp business as a whole.

i. User Behavior

Our graphs showed that there was a higher number of user reviews in the summer months and on weekends (Exhibit 2). This is probably because people are more likely to go out when the weather is more temperate and when they have more leisure time.

ii. Restaurants

Businesses on Yelp belong to multiple categories like ‘Dental’,’Beauty’ and so on which are very different from each other. We found that restaurants had the highest representation in the dataset.Therefore we decided to adopt a focused approach and narrowed our analysis to businesses that fall under the ‘Restaurant’ category. Of all States, Arizona had the highest number of restaurants in the US while Las Vegas had the highest number of restaurants when we compared the cities. (Exhibit 3). Delving more into Las Vegas, we can see that a large number of the restaurants are primarily located in downtown (Exhibit 4)

Using the business category, we separated all of the ethnic restaurants to identify any possible correlation between the number of restaurants and the ethnic breakdown of our top cities. Since we did not have this information, we extracted the numbers from the Census and created a dataset that displays the ethnic breakdown of our top cities. With a scatterplot and a linear regression curve, we see that there is a slight positive correlation between the number of ethnic restaurants and the percentage of their respective ethnic groups (Exhibit 7). From a business context, by looking at the city’s ethnic breakdown, it could help show if there is a strong market for different types of restaurants.

iii. Star Ratings

Star rating have a lot of impact on the restaurant business. People usually take a look at the star ratings first when they were looking for a restaurant on Yelp. In our data, maximum number of restaurants had an average of 4 star rating, followed by 3.5 star rating. (Exhibit 5) Arizona in the US had the highest number of restaurants with 5 star ratings. (Exhibit 6)

3. Sentiment Analysis

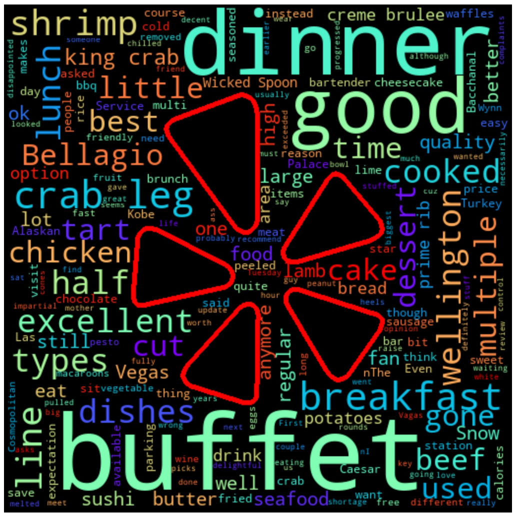
By definition, sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. The capability of understanding customer reviews and manage customer relationship accordingly is crucial to every business. Some of the common uses of sentiment analysis are review monitoring, customer relationship management and intent analysis.

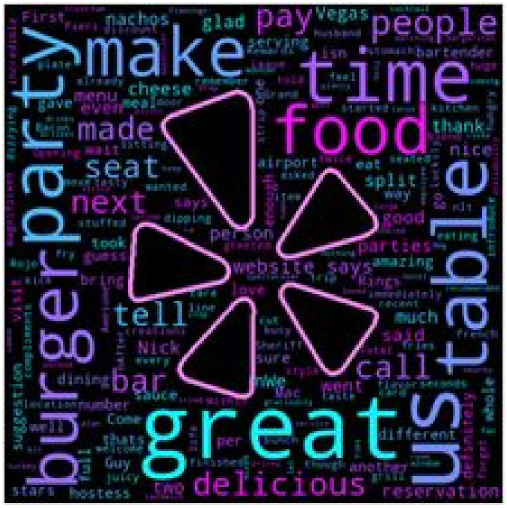
i. Python packages

From our EDA, we discovered that Las Vegas has the most restaurants in the US. To further explore that, we begin our sentiment analysis by comparing two restaurants in Las Vegas. We chose the comparison between “the buffet at Bellagio” and “Guy Fieri’s Vegas Kitchen & Bar”. These two restaurants are comparable because the average ratings and sample size we took are both similar, location and restaurant category are the same. To start, we conducted fundamental text analysis: word tokenizer and word cloud. For our sentiment analysis, we utilized existing Python packages from NLTK. NLTK, or the Natural Language Toolkit, is a series of libraries and programming for natural language processing. It was developed in the Department of Computer and Information Science at the University of Pennsylvania. The famous packages include lexical analysis, n-gram and sentiment analysis packages such as TextBlob and Vader.

a. Word Tokenizer and Word Cloud

Word Tokenizer were used to calculate word count for each review then word clouds were generated to represent the reviews. Word cloud is an excellent way to visualize frequent used words in reviews. The size of each text represents the frequency of occurrence. However, the sentiment could not be fully represented in the word cloud. To do so, we advance our analysis using NLTK packages.



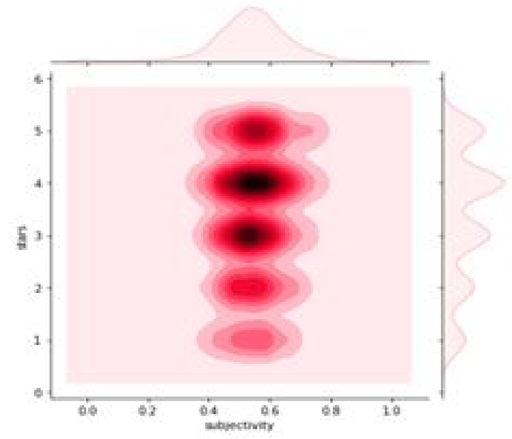


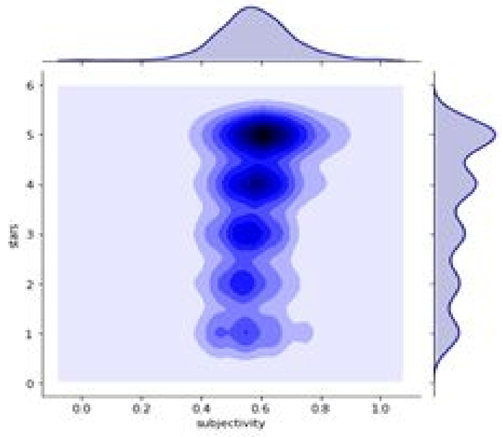
b. TextBlob

The first method we tried on these two restaurants is TextBlob. TextBlob is commonly used to perform basic natural language processing on textual data. Perform sentiment analysis using TextBlob will return a named tuple of the form sentiment (polarity, subjectivity). The polarity score is a float within the range -1.0 (very negative) to 1.0 (very positive). The subjectivity is a float within the range 0.0 to 1.0 where 0.0 is very objective and 1.0 is very subjective.

We evaluate reviews of the restaurants by comparing at their star ratings with polarity and subjectivity.

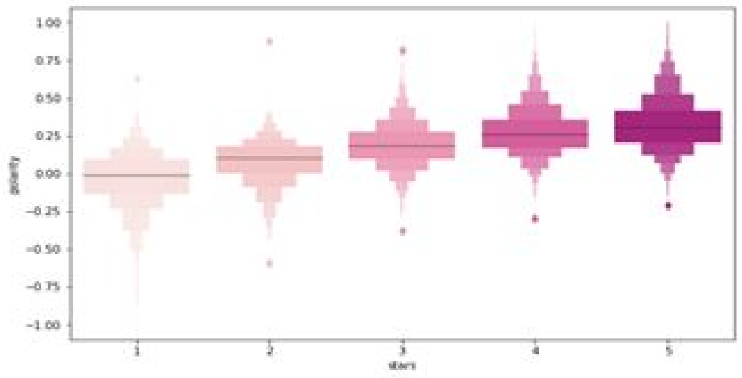
The following density plot shows the relationship between star ratings and subjectivity.

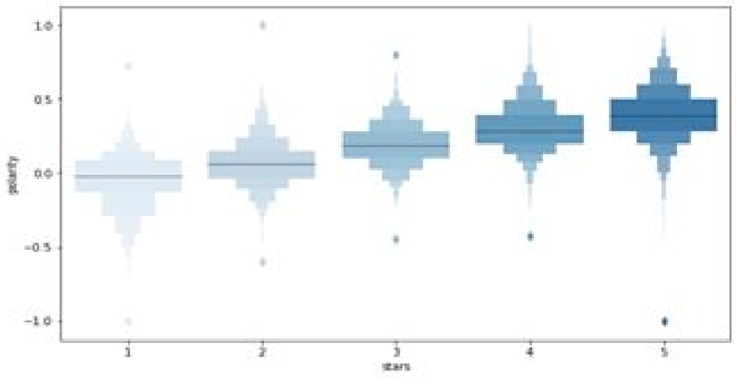




The buffet at Bellagio has more subjective reviews rated 3 and 4 stars, where more 5-star reviews in Guy Fieri's Vegas Kitchen & Bar are subjective. The high subjectivity indicates that customers reviewed their experience based on factors such as food or service. The low subjectivity indicates that customers reviewed their experience based on their emotions. As business, restaurant show look into subjective reviews for improvement.

The following boxplots showed the relationship between star ratings and polarity:





Although the average polarity increases with star ratings, the distribution looks abnormal. Most star ratings have normal distribution of polarity which means some 5-star reviews has low sentiment while 1-star reviews have high sentiment. This is caused by TextBlob algorithm which calculates the average sentiment of each word within the review.

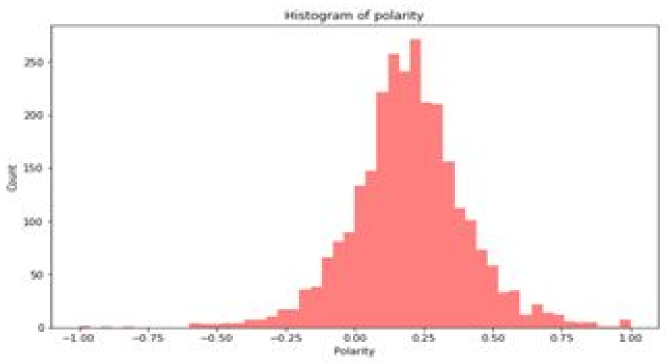
c. Vader

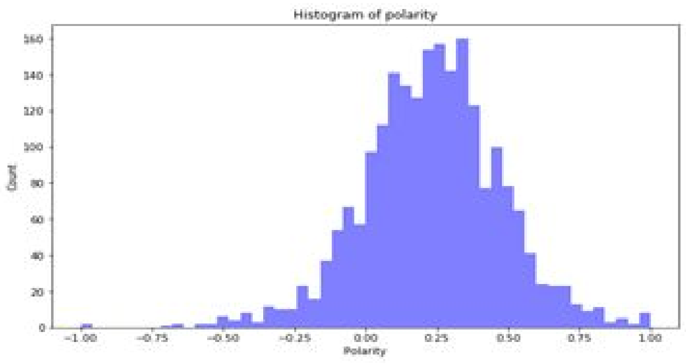
Vader, or Valence Aware Dictionary and sEntiment Reasoner, is a lexicon and rule-based sentiment analysis tool. We utilize Vader as another tool to access sentiment in reviews. Vader returns a dictionary of each sentiment (positive, neutral, negative, compound) ranged from -1.0

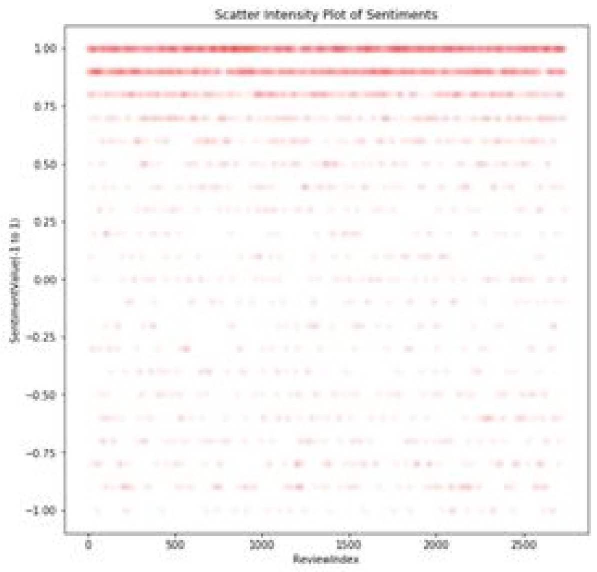
to 1.0. The benefit of Vader is it doesn't only show whether a review is positive or negative, it also shows how much each sentiment is. For example, Vader could return {pos: 0.7163, neu: 0.20, neg: 0.0837, compound: 0.6435} The numbers indicate density of each sentiment. For the purpose of comparison, we will be using compound score.

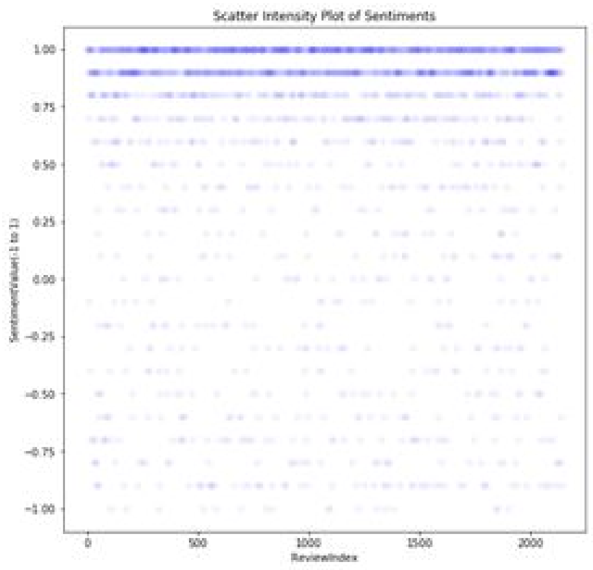
Results:

The following plot shows the sentiment distribution between TextBlob and Vader.









TextBlob shows that most reviews are distributed from "Neutral" to "Positive" while Vader shows more reviews labelled as "Very Positive". The reason for this difference is because TextBlob calculate the average sentiment from each word in a review where Vader calculate the sentiment through the whole review. So, the longer the review is, the less accurate TextBlob predicts. In conclusion, TextBlob is better in predicting shorter text such as social media posts and Vader is better in predicting longer text such as reviews.

ii. Models

a. Logistic Regression

Inspiration

A negative review, especially from an elite user can result in lost opportunity for a business. Businesses usually respond to negative reviews after a certain time but by then the damage is already done. This problem can be solved by performing real time prediction of rating while customer is typing the review so that action can be undertaken even before the review is live! This will significantly improve customer service and reduce the risk of losing customers.

Methodology

Feature Representation

- We used the column ‘reviews’ as the main source of features.
- One set of features were created without removing stop words or performing stemming on them while another set of features was considered after performing these methods. The idea was to assess the impact on the models.
- Moreover, feature engineering was done separately using the following N gram sequences:
  - Unigram – taking one single word at a time
  - Unigram and Bigram – taking pairs of neighboring words along with each word individually

Feature Weighting

The following three kinds of feature weighting were tried:

- Binary - if a word is present in the document, the weight is ‘1’ else ‘0’
- Count - weights equal to number of occurrences of a word
- TFIDF – Term Frequency (TF) Term frequency measures the local importance of the word by taking into account the number of times it appears in a document. Inverse Document Frequency (IDF) takes into account whether the word appears in other documents or not. For a word to be considered a signature word of a document, it shouldn’t appear that often in the other documents. The TF-IDF is the product of these two frequencies. For a word to have high TF-IDF in a document, it must appear a lot of times in said document and must be absent in the other documents.

Logistic Regression Model

A 5% sample (~200K records) was selected from the population using Simple Random Sampling without Replacement. The data was divided into 75% Training data and 25% Testing data. A logistic regression model was then built using these features as the independent variables. The dependent variable that we were trying to predict was the column ‘rating’ which takes the values 1 through 5 on yelp. And thus, this was a multi-classification exercise.
Overall 12 Logistic regression models were built, and the corresponding results were analyzed. (Exhibit 8)

Evaluation

Once the model was built on training data, we used it to predict ratings on testing data. In order to evaluate the performance, we calculated the accuracy in two methods:

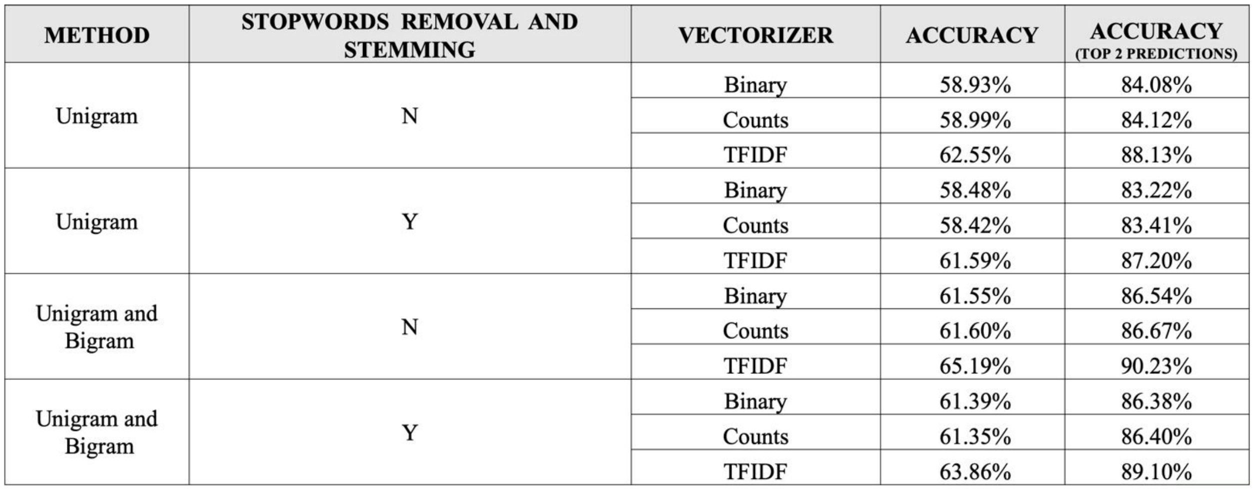
1. Top 1 prediction defined as

$$\frac{\text{Number of times the correct prediction of rating was made}}{\text{Total number of predictions}}$$

2. Top 2 predictions defined as

$$\frac{\text{Number of times the correct rating occurred in top 2 predictions for a review}}{\text{Total number of predictions}}$$

Following table summarizes the results of the 12 models. (The corresponding Confusion Matrices can be found in the Exhibit 9 & 10)

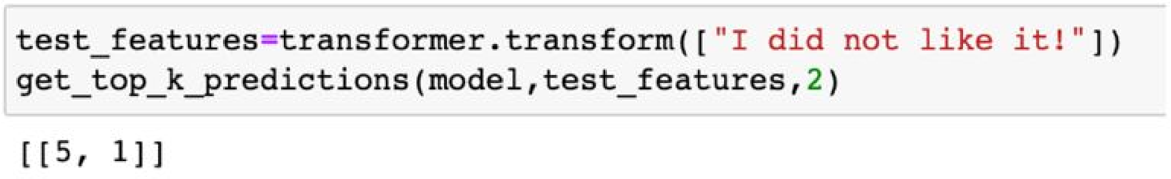


METHOD	STOPWORDS REMOVAL AND STEMMING	VECTORIZER	ACCURACY	ACCURACY (TOP 2 PREDICTIONS)
Unigram	N	Binary	58.93%	84.08%
		Counts	58.99%	84.12%
		TFIDF	62.55%	88.13%
Unigram	Y	Binary	58.48%	83.22%
		Counts	58.42%	83.41%
		TFIDF	61.59%	87.20%
Unigram and Bigram	N	Binary	61.55%	86.54%
		Counts	61.60%	86.67%
		TFIDF	65.19%	90.23%
Unigram and Bigram	Y	Binary	61.39%	86.38%
		Counts	61.35%	86.40%
		TFIDF	63.86%	89.10%

Validation on new text

In addition to predicting ratings using these models on the testing data, we also tested them on new texts to compare the predicted ratings. The following were the results obtained from the three models which were built after removal of stop words and performing stemming and using features in unigram sequence

BINARY

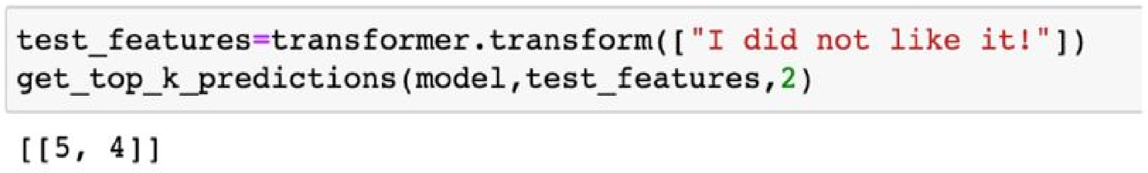


```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)

[[5, 1]]
```

The model does not predict the correct rating in the first spot but makes the right prediction in the second spot. Interestingly, the top two predictions made by this model are in the opposite ends of the rating scale.

COUNTS

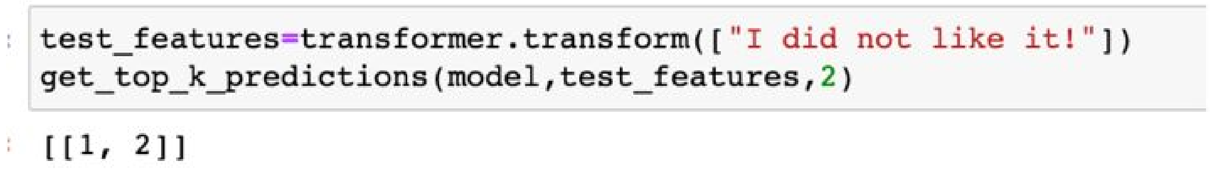


```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)

[[5, 4]]
```

The model does not predict the correct rating in either of the two spots. Both the predictions are on the higher end of the rating scale while the text ideally belongs in the lower end.

TFIDF



```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)

[[1, 2]]
```

The model predicts the correct rating in the first spot. Moreover, the prediction in the second spot also makes logical sense. (Similar comparisons from all the models can be found in Exhibit 11& 12)

b. Random Forest

We first created a label for the reviews using two approaches based on the star ratings of the reviews. We divided the reviews as -

1. Positive and Negative or 1 and 0 with
    a. Rating 1,2,3 as 0

b. Rating 4,5 as 1
2. Positive, Neutral and Negative or 1, 0, -1.
   a. Rating 1,2 as -1
   b. Ration 3 as 0
   c. Rating 4,5 as 1

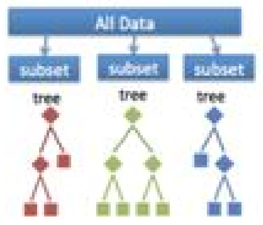
After this label creation, we then used TFIDF and Bag of Words for feature engineering.
We then used two classification models - Random Forest and Support Vector Machine to predict the sentiment of the restaurant reviews based on two approaches in each of the models such as:

1. Keeping all the numbers, punctuation marks and hashtags.
2. Removing all the numbers, punctuation marks and hashtags.

The following table summarizes the results of our modelling process:
(Confusion Matrix Exhibit 13 & 14)

METHOD	VECTORIZER	ACCURACY (Approach 1)	ACCURACY (Approach 2)
Random Forest	BOW	77.72%	61.91%
	TFIDF	78.04%	61.38%
SVM	BOW	87.45%	74.25%
	TFIDF	86.43%	73.21%
Random Forest	BOW	78.30%	61.24%
	TFIDF	78.52%	61.64%
SVM	BOW	86.68%	74.71%
	TFIDF	86.46%	74.65%

Random Forest Model:
Random Forest is a supervised machine learning algorithm used for classification.
It is a combination of multiple decision trees. The output of the random forest algorithm is the weighted accuracy of all the decision trees which it uses to predict the dependent variable.



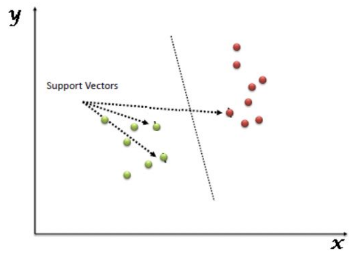
After using Random Forest algorithm, the best model that we got considering all the approaches had an accuracy of 78.30% and produced a confusion matrix as follows: -

	0	1
0	6388	1112
1	2142	5358

This model had the lowest number of False Positives out of all the random forest models that we built.

c. SVM

Support Vector Machine:
In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features we had) with the value of each feature being the value of a particular coordinate. Then, we performed classification by finding the hyper-plane that differentiate the two classes very well.

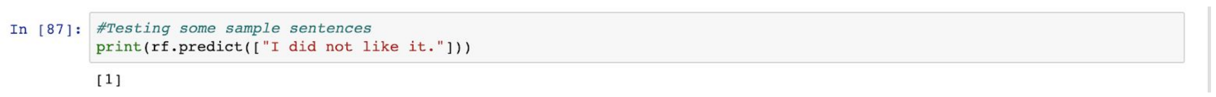


After using Support Vector Machine algorithm, the best model that we got considering all the approaches had an accuracy of 86.83% and produced a confusion matrix as follows:
(Confusion Matrix Exhibit 15& 16)

	0	1
0	6564	936
1	1039	6461

Validation on new text

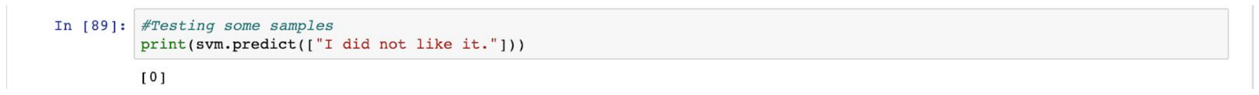
We used the same sentence “I did not like it” to test whether our models are able to predict the right rating.
Random forest model produced a false positive as it predicted a negative review as a positive review.



```
In [87]: #Testing some sample sentences
         print(rf.predict(["I did not like it."]))

         [1]
```

Whereas, the SVM model produced a True Negative value as it correctly predicted a negative review as a negative review.



```
In [89]: #Testing some samples
         print(svm.predict(["I did not like it."]))

         [0]
```

The reason we shortlisted these models was because we wanted to reduce the number of reviews which were actually negative but were classified as positive. This is because a False Positive would mean that the restaurant would consider a negative review as positive and lose out on any potential opportunities for improvement.

4. Learning and Insights

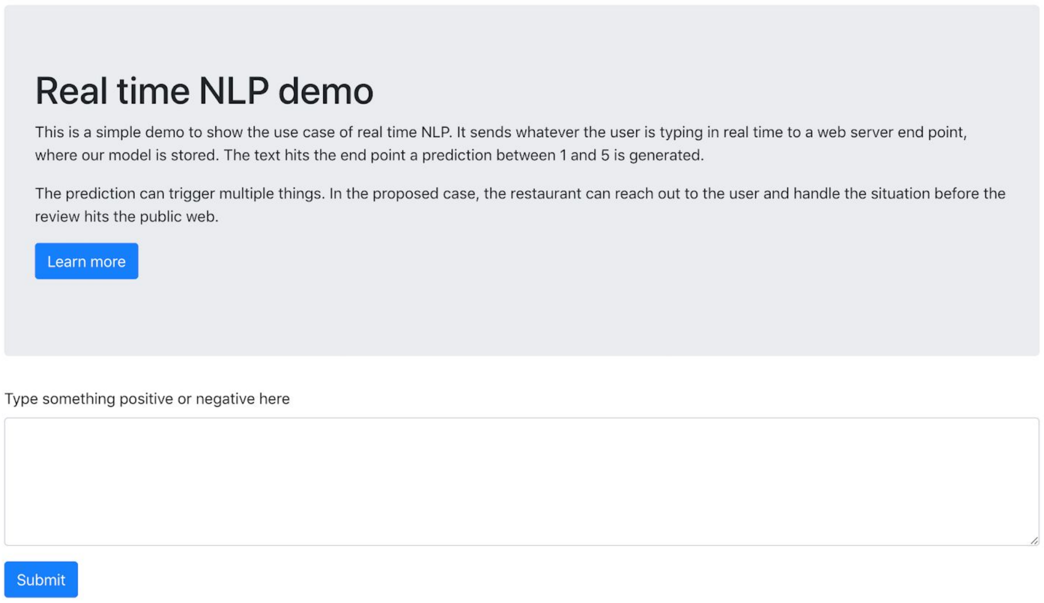
- TF-IDF based weighting performs better than Binary, Count based schemes as well as Bag of Words approach
- We can do feature engineering using N gram sequence as the richer the text field, the better the overall performance of the classifier
- Accuracy is not the only benchmark and context plays a vital role
- The algorithm cannot understand the tone of the review and detect human emotions like irony, sarcasm, jokes or exaggerations and failure to recognize these can skew the results
- Businesses can use this methodology to predict rating in real-time and undertake relevant actions to improve customer service
- For future work, we can create new text fields by adding the content from ‘Tips’ and analyzing how our models perfom on addition of this new data

5. Demo

One practical application of Sentiment Analysis would be real-time prediction of rating when a customer is adding his/her review. This way if a low rating gets predicted, the restaurant can immediately offer some assistance to improve the customer experience. An action would be taken even before the review gets published.

We built a prototype of this application to demonstrate this idea. As we saw in the presentation, the message pops up while the customer is typing a review. The below screenshots capture the examples covered in the presentation.

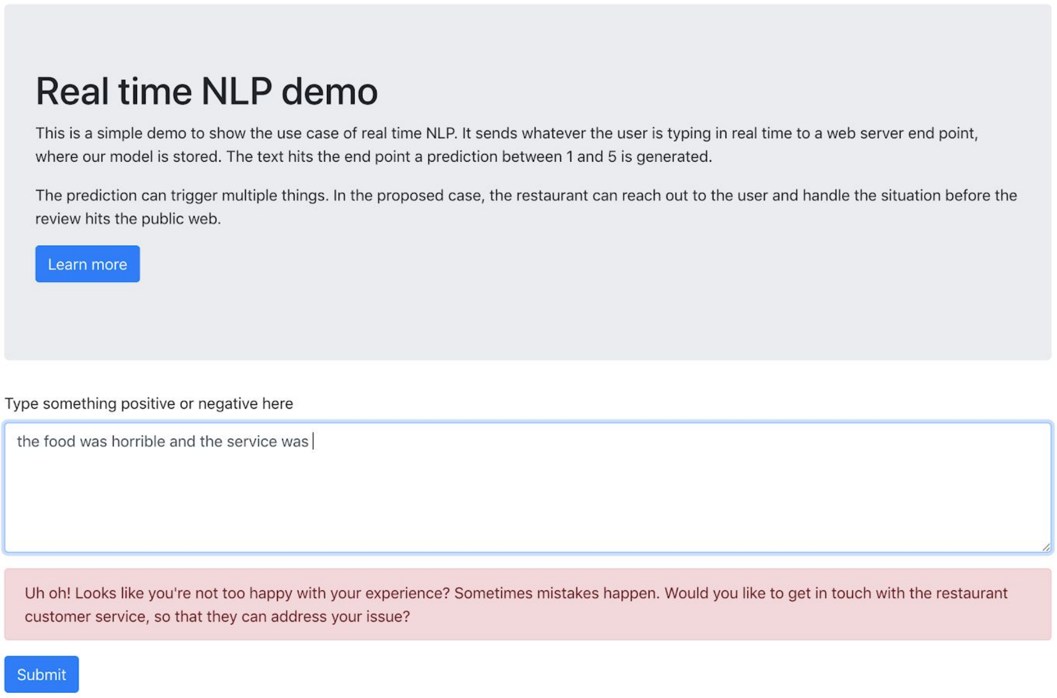
- The main page of the prototype:


Real time NLP demo

This is a simple demo to show the use case of real time NLP. It sends whatever the user is typing in real time to a web server end point, where our model is stored. The text hits the end point a prediction between 1 and 5 is generated.

The prediction can trigger multiple things. In the proposed case, the restaurant can reach out to the user and handle the situation before the review hits the public web.

- A dialog box pops up even before the user has finished writing the review. Since the model predicted that the rating of this particular review is going to be low and thus this is a negative review, the resultant message asks whether the user would like to talk to the customer representative.


Real time NLP demo

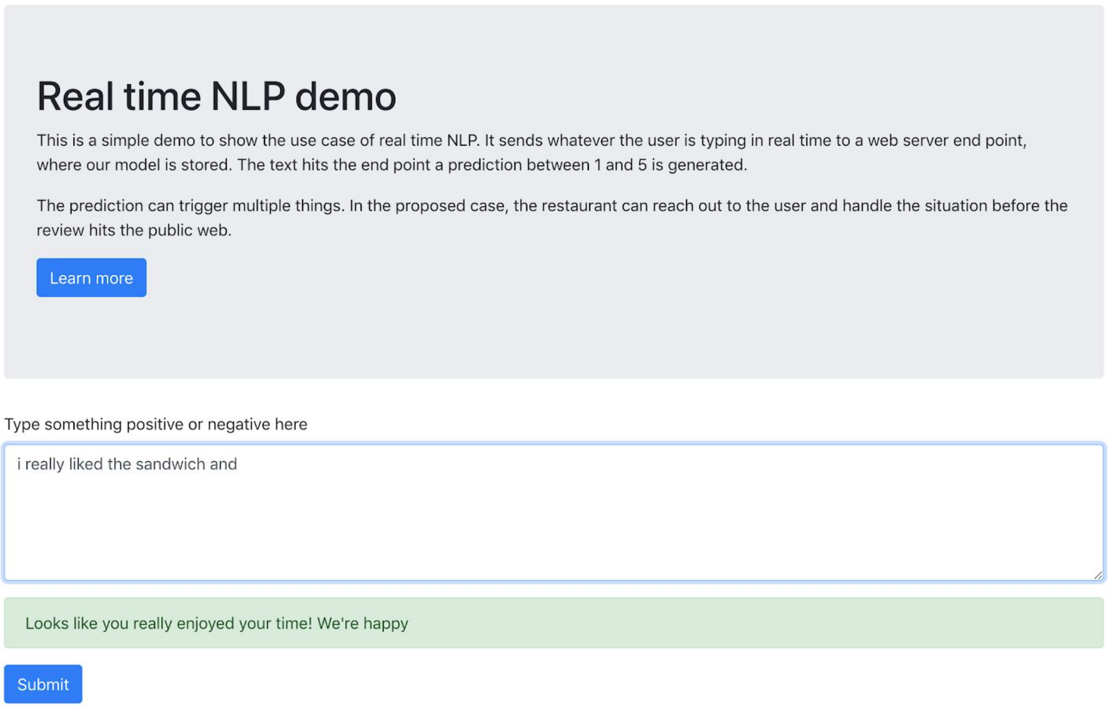
This is a simple demo to show the use case of real time NLP. It sends whatever the user is typing in real time to a web server end point, where our model is stored. The text hits the end point a prediction between 1 and 5 is generated.

The prediction can trigger multiple things. In the proposed case, the restaurant can reach out to the user and handle the situation before the review hits the public web.

the food was horrible and the service was

Uh oh! Looks like you're not too happy with your experience? Sometimes mistakes happen. Would you like to get in touch with the restaurant customer service, so that they can address your issue?

- Since the following sentence is a positive review the tool simply shares a more light hearted message. Notice that even here the message is shown before the text is complete.



Real time NLP demo

This is a simple demo to show the use case of real time NLP. It sends whatever the user is typing in real time to a web server end point, where our model is stored. The text hits the end point a prediction between 1 and 5 is generated.

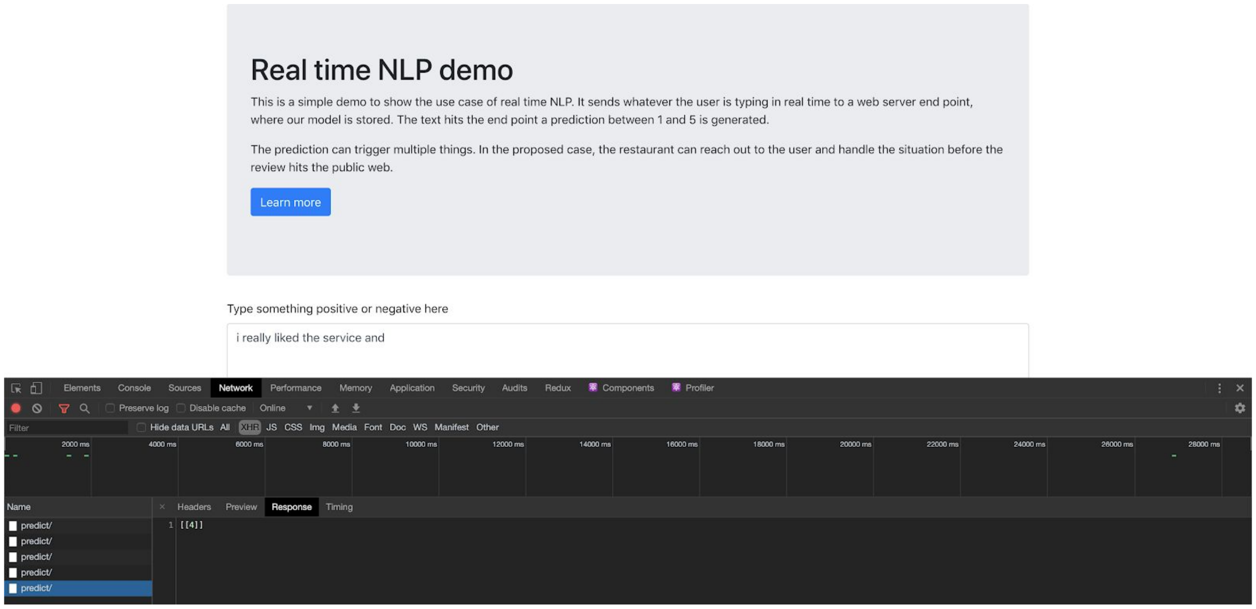
The prediction can trigger multiple things. In the proposed case, the restaurant can reach out to the user and handle the situation before the review hits the public web.

Type something positive or negative here

i really liked the sandwich and

Looks like you really enjoyed your time! We're happy

- In terms of operations in the backend, this is how the mechanism looks like. While the customer is typing a review, the server sends a prediction request to the model which in turn sends a predicted customer rating as an output. On the basis of different values of the ratings, different kinds of messages get displayed to the user. The core idea being real time prediction and action.



Real time NLP demo

This is a simple demo to show the use case of real time NLP. It sends whatever the user is typing in real time to a web server end point, where our model is stored. The text hits the end point a prediction between 1 and 5 is generated.

The prediction can trigger multiple things. In the proposed case, the restaurant can reach out to the user and handle the situation before the review hits the public web.

Type something positive or negative here

i really liked the service and

```
Name
predict/
predict/
predict/
predict/
predict/
```

```
[[4]]
```

6. Appendix

Exhibit 1: Yelp's Growth

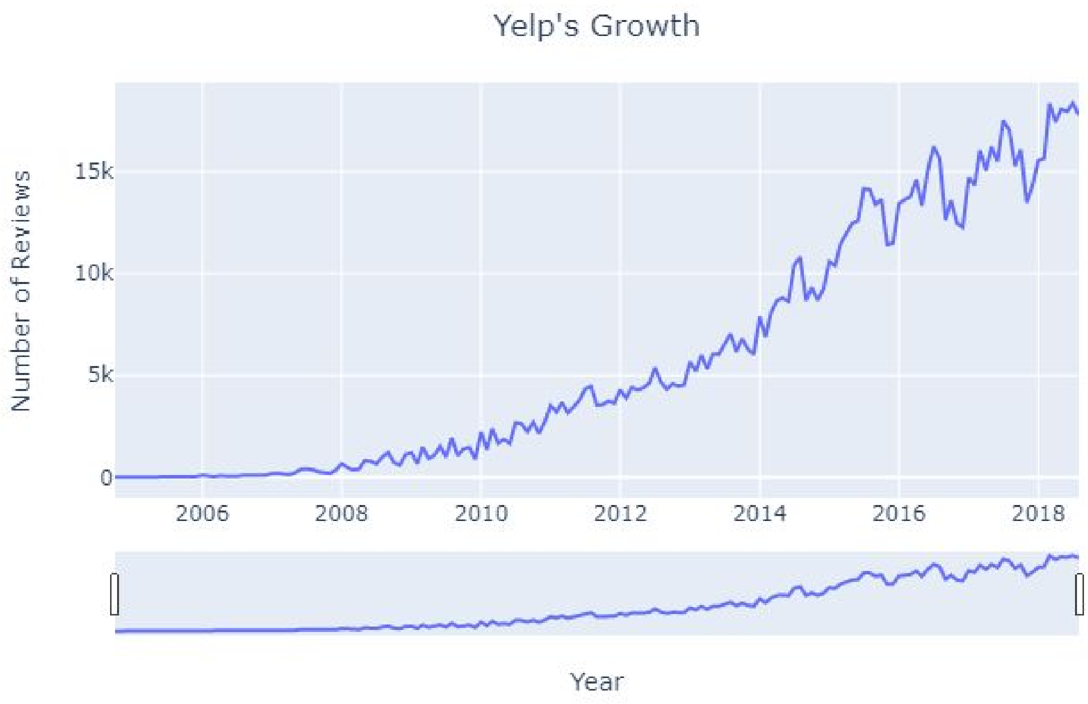
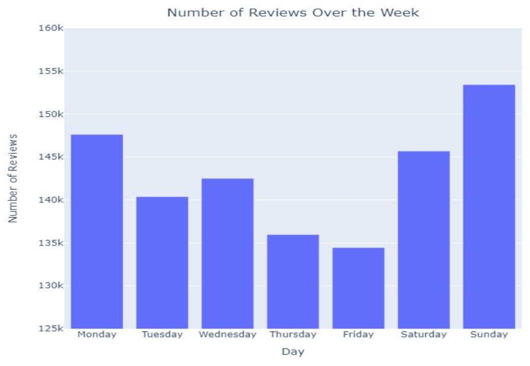


Exhibit 2: User Behavior



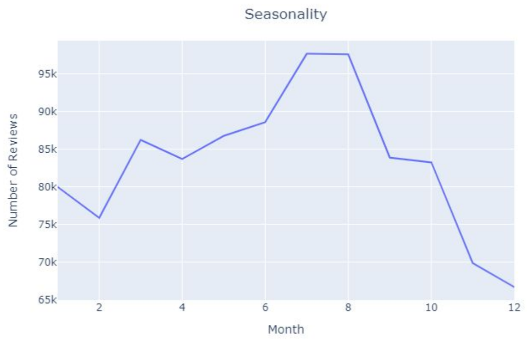
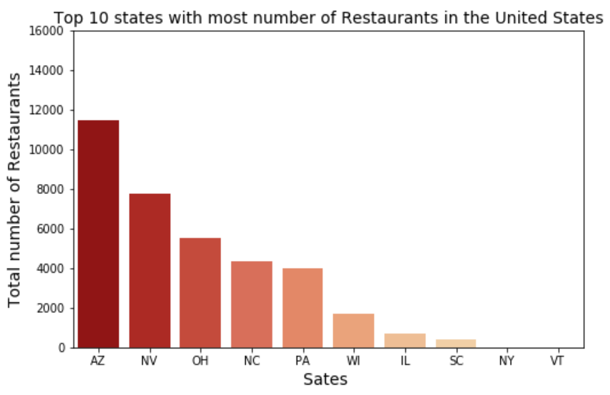


Exhibit 3: Restaurant Overview



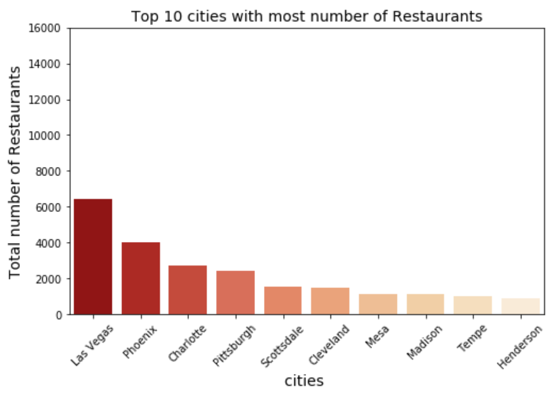
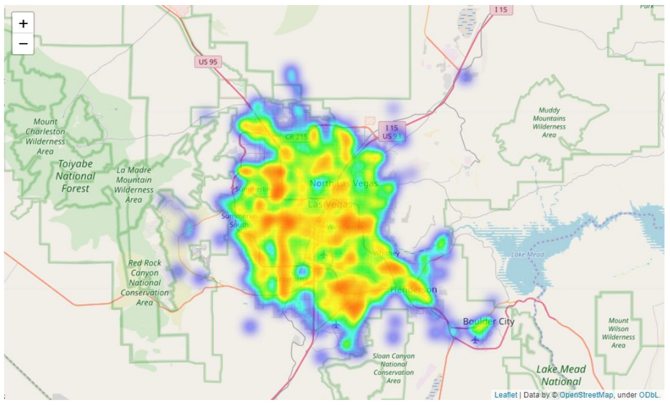


Exhibit 4: Las Vegas Heatmap



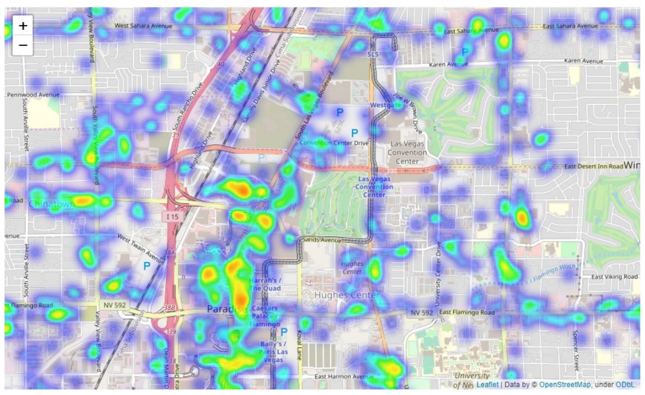


Exhibit 5: Star Rating Distribution

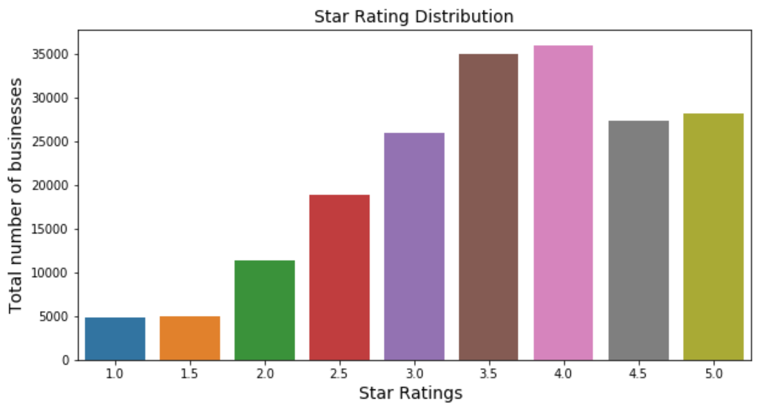


Exhibit 6: Star Rating Distribution

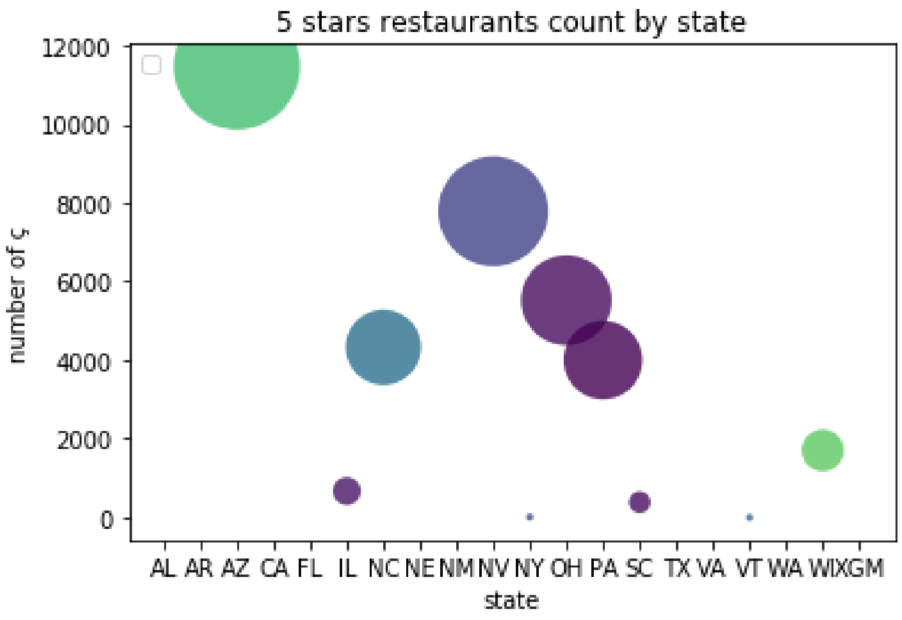


Exhibit 7: Ethnic Breakdown

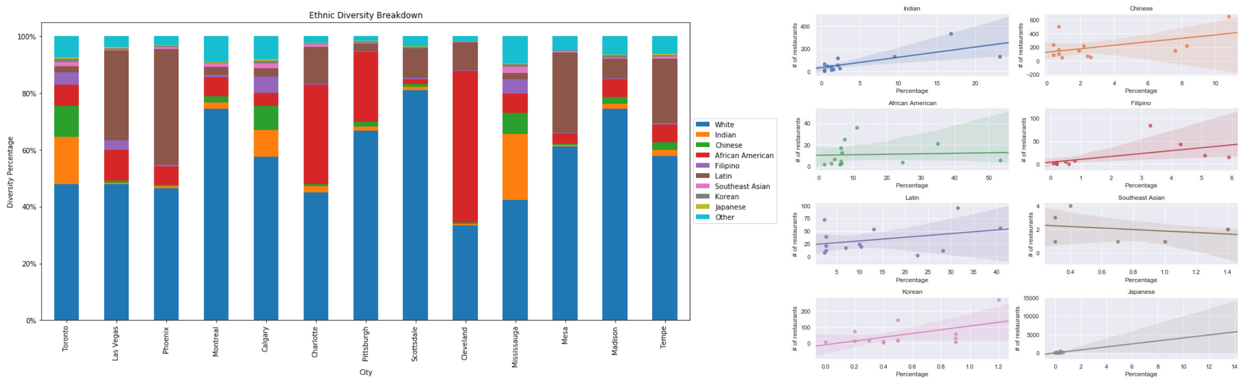


Exhibit 8: Logistic Regression Models

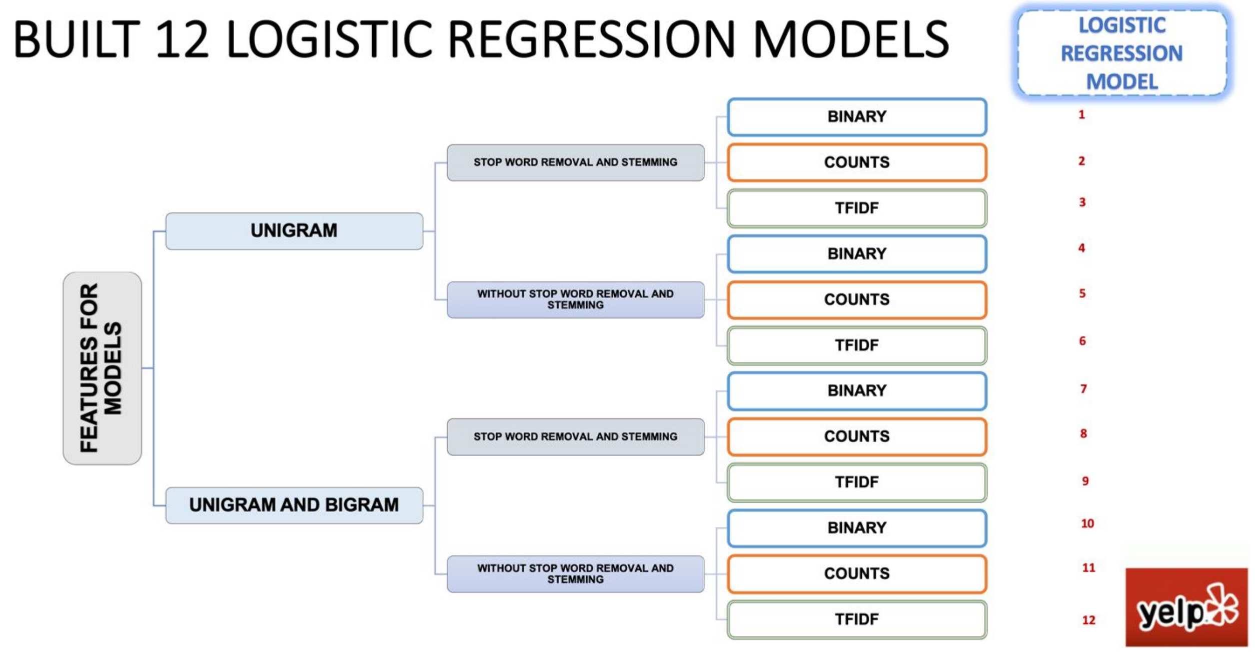
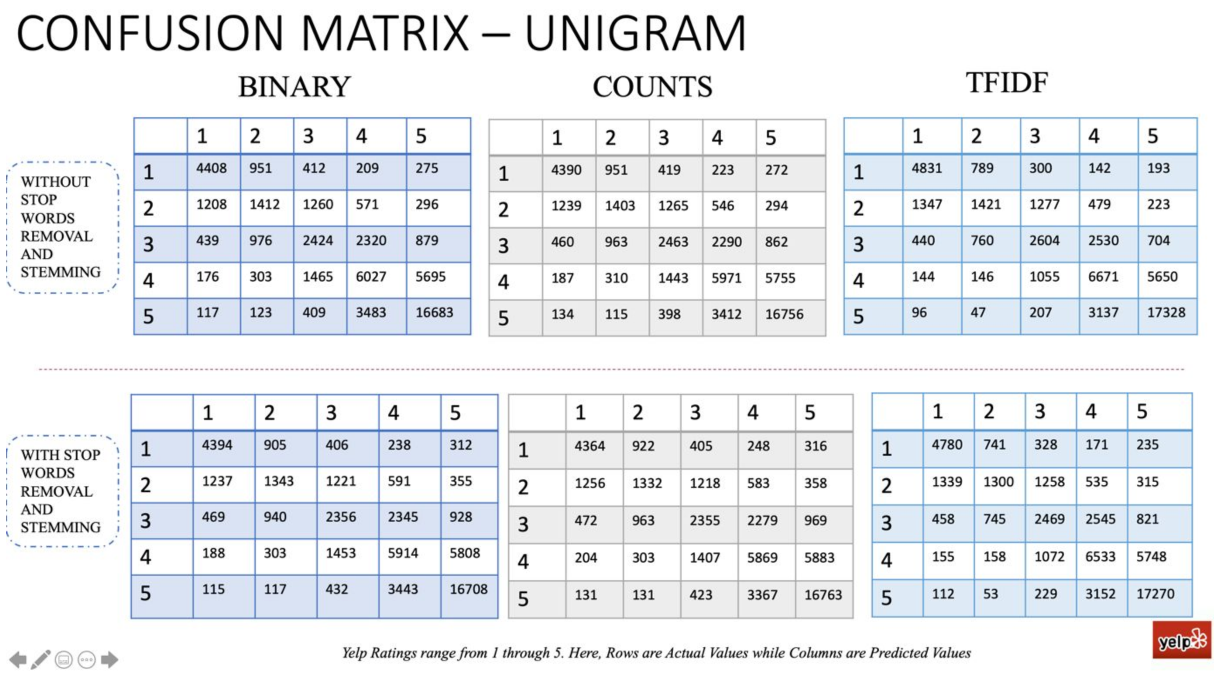

BUILT 12 LOGISTIC REGRESSION MODELS
LOGISTIC REGRESSION MODEL
FEATURES FOR MODELS
UNIGRAM
UNIGRAM AND BIGRAM
STOP WORD REMOVAL AND STEMMING
WITHOUT STOP WORD REMOVAL AND STEMMING
STOP WORD REMOVAL AND STEMMING
WITHOUT STOP WORD REMOVAL AND STEMMING
BINARY
COUNTS
TFIDF
BINARY
COUNTS
TFIDF
BINARY
COUNTS
TFIDF
BINARY
COUNTS
TFIDF
1
2
3
4
5
6
7
8
9
10
11
12
yelp

Exhibit 9 & 10 : Logistic Regression Confusion Matrix



CONFUSION MATRIX – UNIGRAM

WITHOUT STOP WORDS REMOVAL AND STEMMING

BINARY

	1	2	3	4	5
1	4408	951	412	209	275
2	1208	1412	1260	571	296
3	439	976	2424	2320	879
4	176	303	1465	6027	5695
5	117	123	409	3483	16683

COUNTS

	1	2	3	4	5
1	4390	951	419	223	272
2	1239	1403	1265	546	294
3	460	963	2463	2290	862
4	187	310	1443	5971	5755
5	134	115	398	3412	16756

TFIDF

	1	2	3	4	5
1	4831	789	300	142	193
2	1347	1421	1277	479	223
3	440	760	2604	2530	704
4	144	146	1055	6671	5650
5	96	47	207	3137	17328

WITH STOP WORDS REMOVAL AND STEMMING

BINARY

	1	2	3	4	5
1	4394	905	406	238	312
2	1237	1343	1221	591	355
3	469	940	2356	2345	928
4	188	303	1453	5914	5808
5	115	117	432	3443	16708

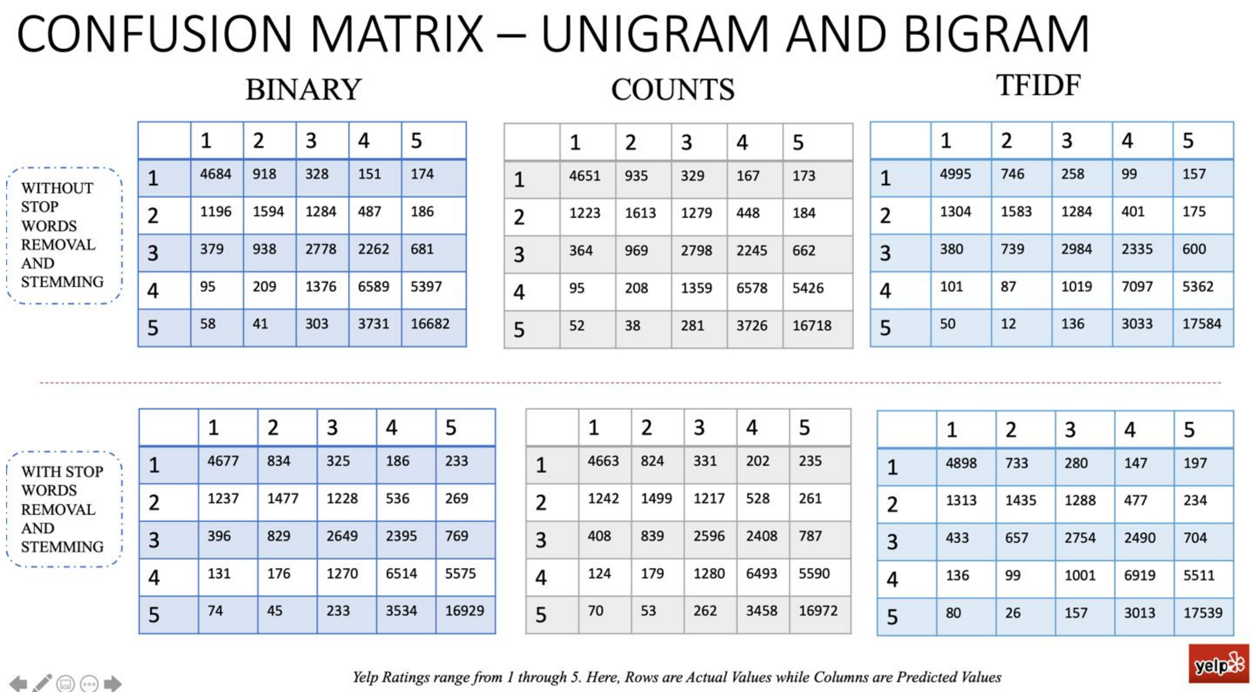
COUNTS

	1	2	3	4	5
1	4364	922	405	248	316
2	1256	1332	1218	583	358
3	472	963	2355	2279	969
4	204	303	1407	5869	5883
5	131	131	423	3367	16763

TFIDF

	1	2	3	4	5
1	4780	741	328	171	235
2	1339	1300	1258	535	315
3	458	745	2469	2545	821
4	155	158	1072	6533	5748
5	112	53	229	3152	17270

Yelp Ratings range from 1 through 5. Here, Rows are Actual Values while Columns are Predicted Values



CONFUSION MATRIX – UNIGRAM AND BIGRAM

WITHOUT STOP WORDS REMOVAL AND STEMMING

BINARY

	1	2	3	4	5
1	4684	918	328	151	174
2	1196	1594	1284	487	186
3	379	938	2778	2262	681
4	95	209	1376	6589	5397
5	58	41	303	3731	16682

COUNTS

	1	2	3	4	5
1	4651	935	329	167	173
2	1223	1613	1279	448	184
3	364	969	2798	2245	662
4	95	208	1359	6578	5426
5	52	38	281	3726	16718

TFIDF

	1	2	3	4	5
1	4995	746	258	99	157
2	1304	1583	1284	401	175
3	380	739	2984	2335	600
4	101	87	1019	7097	5362
5	50	12	136	3033	17584

WITH STOP WORDS REMOVAL AND STEMMING

BINARY

	1	2	3	4	5
1	4677	834	325	186	233
2	1237	1477	1228	536	269
3	396	829	2649	2395	769
4	131	176	1270	6514	5575
5	74	45	233	3534	16929

COUNTS

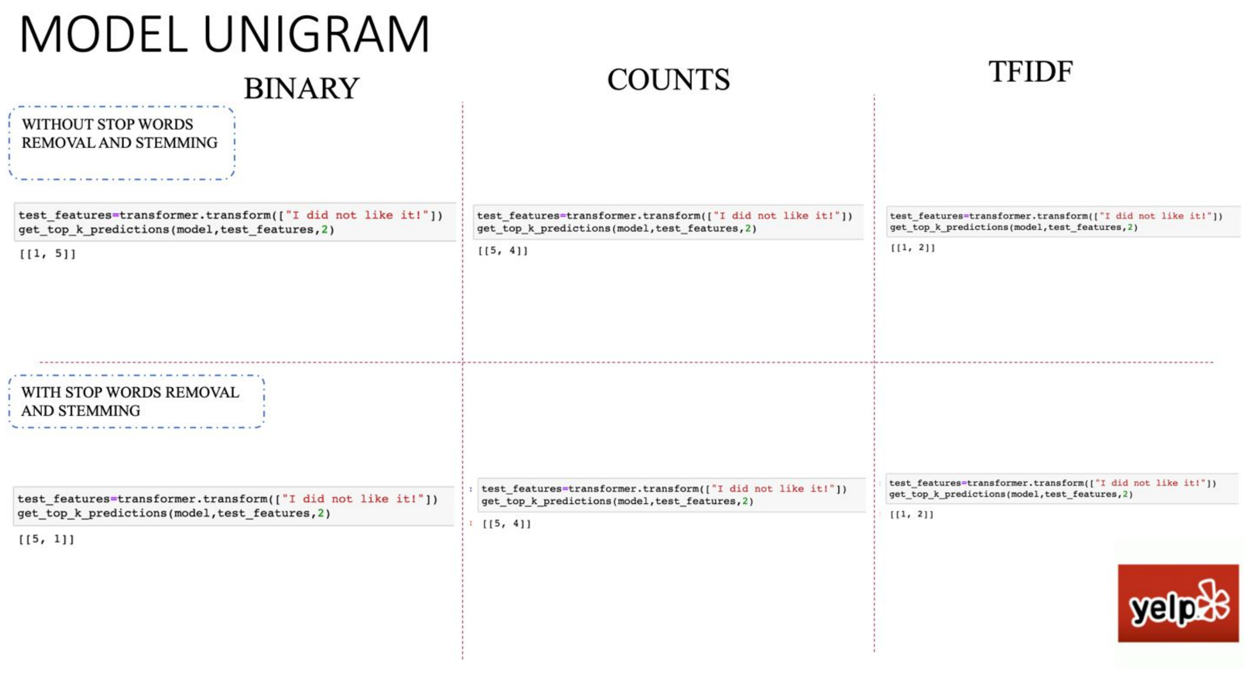
	1	2	3	4	5
1	4663	824	331	202	235
2	1242	1499	1217	528	261
3	408	839	2596	2408	787
4	124	179	1280	6493	5590
5	70	53	262	3458	16972

TFIDF

	1	2	3	4	5
1	4898	733	280	147	197
2	1313	1435	1288	477	234
3	433	657	2754	2490	704
4	136	99	1001	6919	5511
5	80	26	157	3013	17539

Yelp Ratings range from 1 through 5. Here, Rows are Actual Values while Columns are Predicted Values

Exhibit 11&12: Logistic Regression Model Comparison



MODEL UNIGRAM

WITHOUT STOP WORDS REMOVAL AND STEMMING

BINARY

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[1, 5]]
```

COUNTS

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[5, 4]]
```

TFIDF

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[1, 2]]
```

WITH STOP WORDS REMOVAL AND STEMMING

BINARY

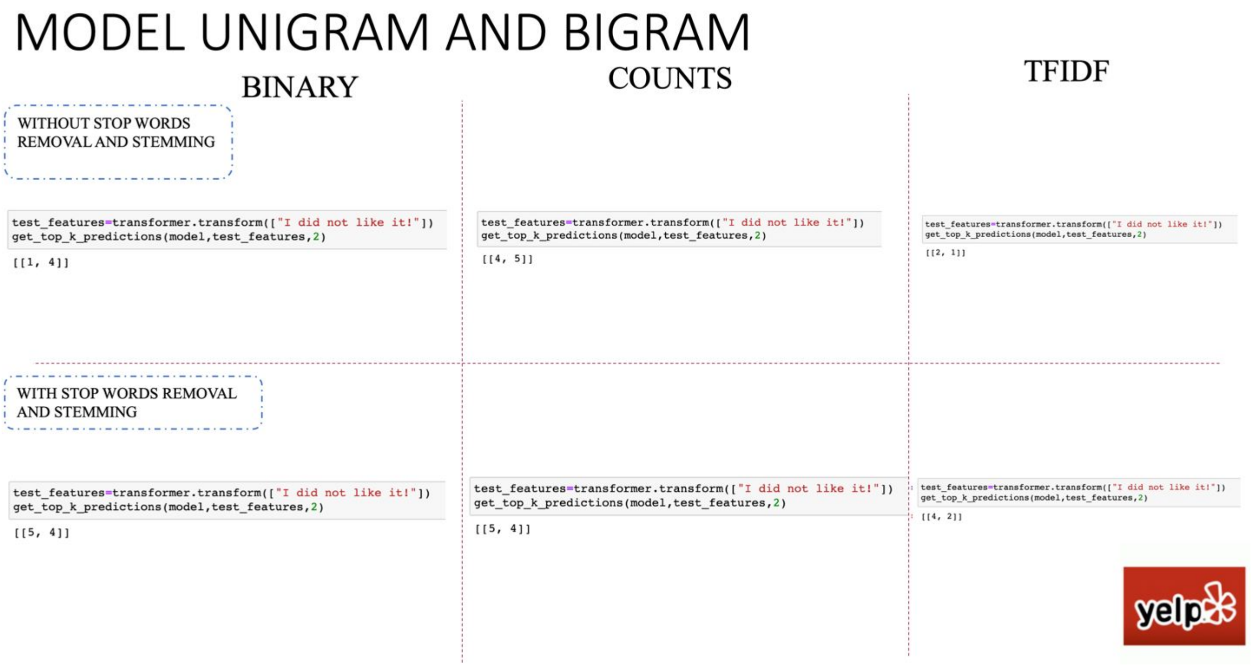
```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[5, 1]]
```

COUNTS

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[5, 4]]
```

TFIDF

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[1, 2]]
```



MODEL UNIGRAM AND BIGRAM

BINARY — COUNTS — TFIDF

WITHOUT STOP WORDS REMOVAL AND STEMMING

BINARY:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[1, 4]]
```

COUNTS:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[4, 5]]
```

TFIDF:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[2, 1]]
```

WITH STOP WORDS REMOVAL AND STEMMING

BINARY:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[5, 4]]
```

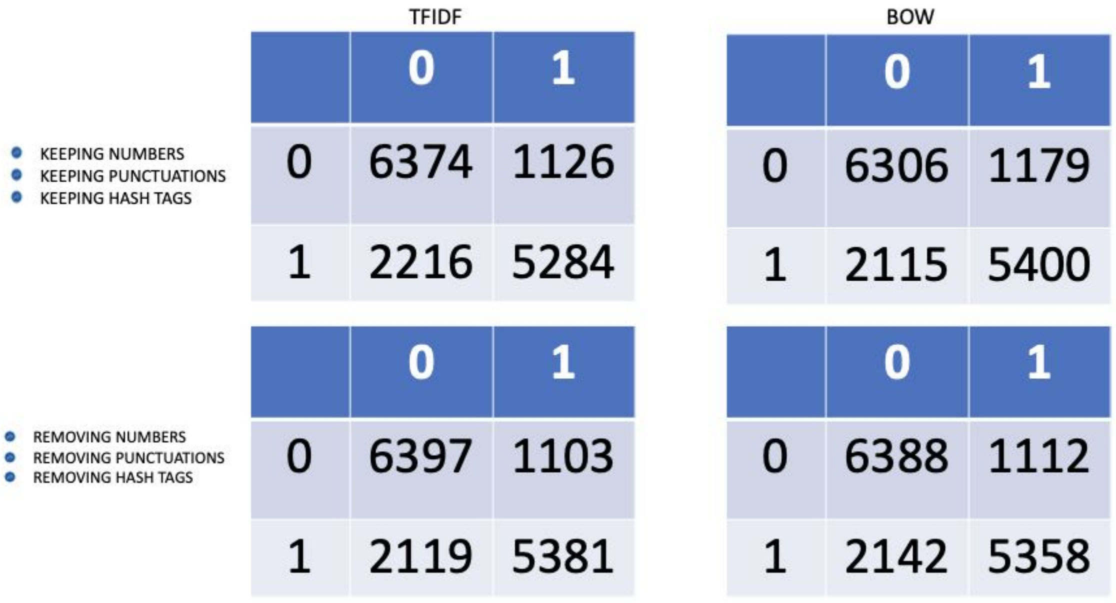
COUNTS:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[5, 4]]
```

TFIDF:

```
test_features=transformer.transform(["I did not like it!"])
get_top_k_predictions(model,test_features,2)
[[4, 2]]
```

Exhibit 13 &14: Random Forest Confusion Matrix



- KEEPING NUMBERS
- KEEPING PUNCTUATIONS
- KEEPING HASH TAGS

TFIDF

	0	1
0	6374	1126
1	2216	5284

BOW

	0	1
0	6306	1179
1	2115	5400

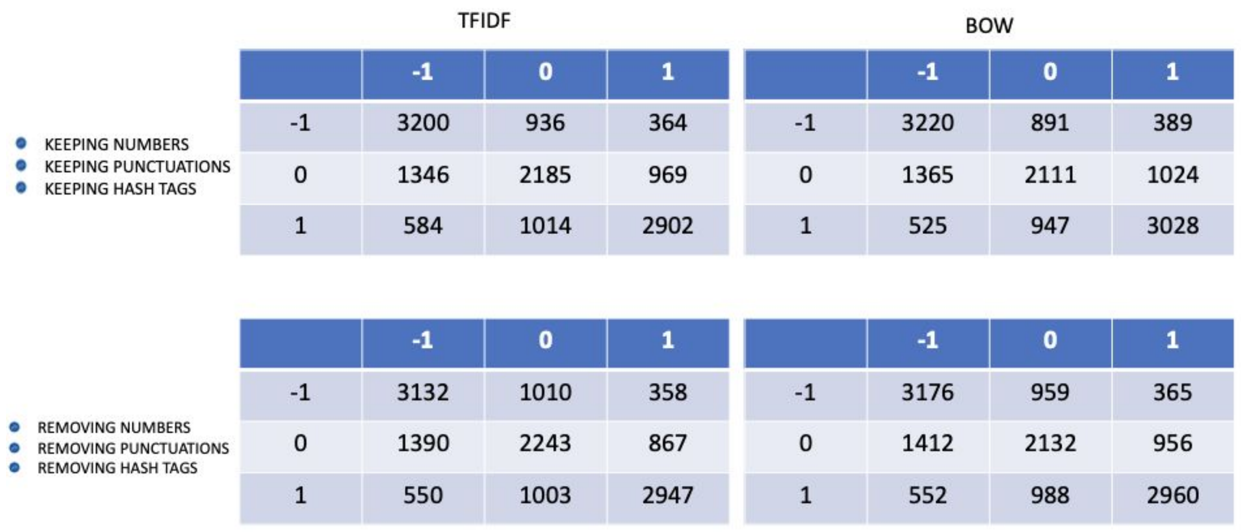
- REMOVING NUMBERS
- REMOVING PUNCTUATIONS
- REMOVING HASH TAGS

TFIDF

	0	1
0	6397	1103
1	2119	5381

BOW

	0	1
0	6388	1112
1	2142	5358



- KEEPING NUMBERS
- KEEPING PUNCTUATIONS
- KEEPING HASH TAGS

TFIDF

	-1	0	1
-1	3200	936	364
0	1346	2185	969
1	584	1014	2902

BOW

	-1	0	1
-1	3220	891	389
0	1365	2111	1024
1	525	947	3028

- REMOVING NUMBERS
- REMOVING PUNCTUATIONS
- REMOVING HASH TAGS

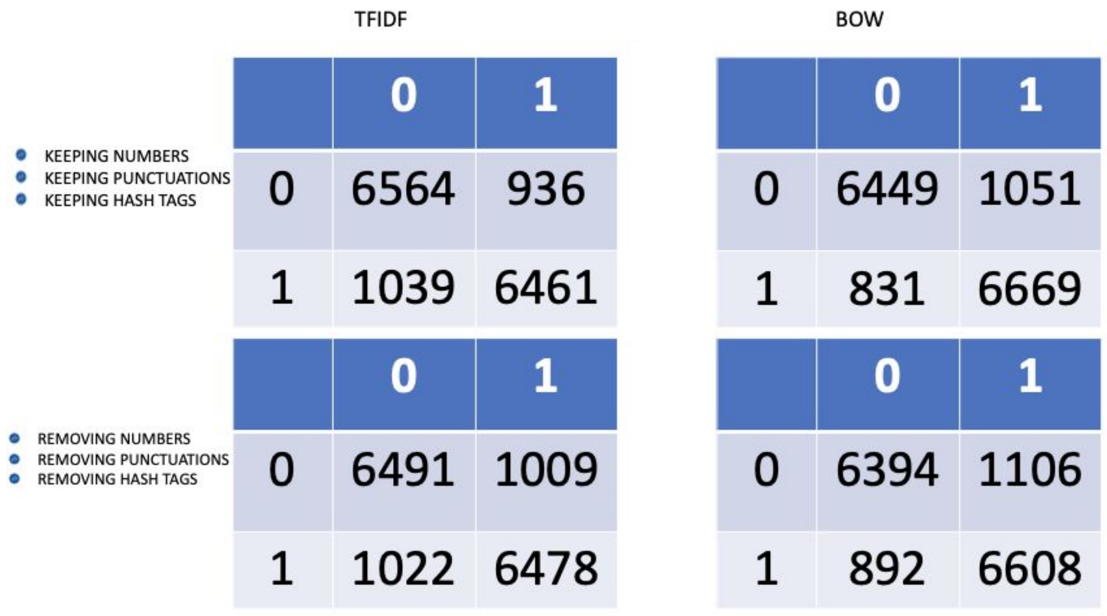
TFIDF

	-1	0	1
-1	3132	1010	358
0	1390	2243	867
1	550	1003	2947

BOW

	-1	0	1
-1	3176	959	365
0	1412	2132	956
1	552	988	2960

Exhibit 15 &16: Support Vector Machine Confusion Matrix



- KEEPING NUMBERS
- KEEPING PUNCTUATIONS
- KEEPING HASH TAGS

TFIDF

	0	1
0	6564	936
1	1039	6461

BOW

	0	1
0	6449	1051
1	831	6669

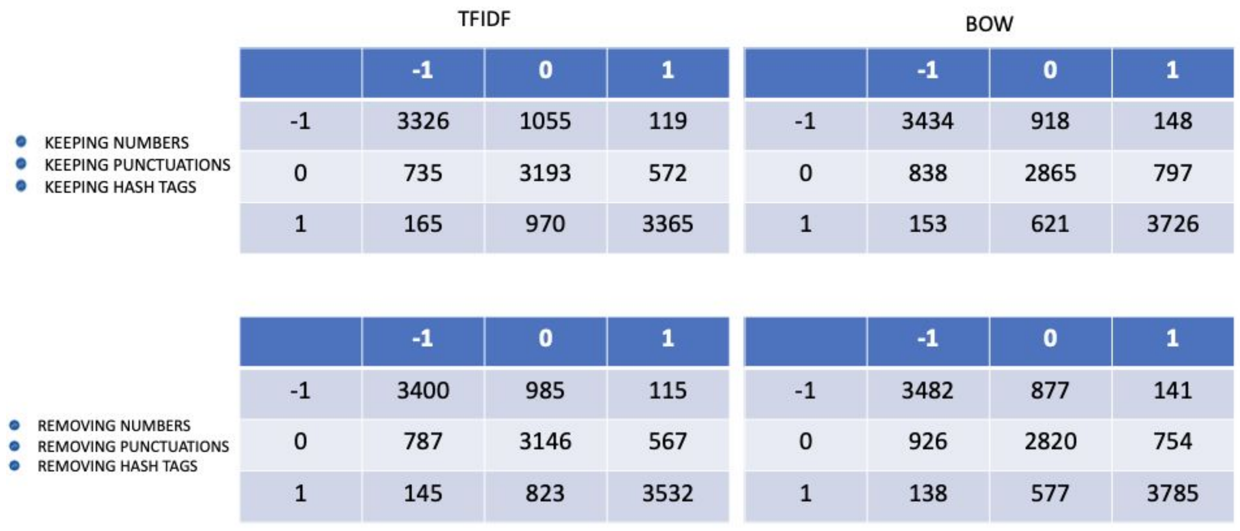
- REMOVING NUMBERS
- REMOVING PUNCTUATIONS
- REMOVING HASH TAGS

TFIDF

	0	1
0	6491	1009
1	1022	6478

BOW

	0	1
0	6394	1106
1	892	6608



TFIDF / BOW

- KEEPING NUMBERS
- KEEPING PUNCTUATIONS
- KEEPING HASH TAGS

TFIDF

	-1	0	1
-1	3326	1055	119
0	735	3193	572
1	165	970	3365

BOW

	-1	0	1
-1	3434	918	148
0	838	2865	797
1	153	621	3726

- REMOVING NUMBERS
- REMOVING PUNCTUATIONS
- REMOVING HASH TAGS

TFIDF

	-1	0	1
-1	3400	985	115
0	787	3146	567
1	145	823	3532

BOW

	-1	0	1
-1	3482	877	141
0	926	2820	754
1	138	577	3785

7. Sources

1. https://www.washingtonpost.com/business/amys-baking-co-meltdown-begs-the-question-is-yelp-bad-for-small-business/2013/05/17/9c07e1aa-beff-11e2-89c9-3be8095fe767_story.html
2. https://www.quora.com/How-much-does-one-bad-Yelp-review-affect-a-local-business
3. https://www.washingtonpost.com/business/technology/how-much-power-does-a-yelp-review-have/2013/05/22/6b8f961c-c210-11e2-8c3b-0b5e9247e8ca_story.html
4. https://hbswk.hbs.edu/item/the-yelp-factor-are-consumer-reviews-good-for-business
5. https://web.stanford.edu/~jurafsky/slp3/5.pdf
6. https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python/
7. http://www12.statcan.ca/census-recensement/2016/as-sa/fogs-spg/Facts-CSD-eng.cfm?TOPIC=7&LANG=eng&GK=CSD&GC=3520005
8. https://datausa.io/profile/geo/las-vegas-nv/
9. http://www.usa.com/las-vegas-nv-population-and-races.htm