



Leslie Pham

Problem Statement

Determine the importance of certain features in a business's success as well as evaluate sentiment in reviews of Las Vegas businesses.

Data:

- User Information
- Business Information
- Tips (text)
- Reviews
- Check-in Details

Analysis and Modeling Approach

- Natural Language Processing
- Sentiment Analysis
- Visual Analysis
- Feature Importance
- Topic Modeling

Technical Appendix

Packages used:

- Pandas
- NumPy
- Seaborn
- RegEx
- Natural Language Tool Kit
- SciKitLearn
- Matplotlib
- Textblob - Naive Bayes Classifier

Goals and Success Metrics

- A model that is somewhat accurate in identifying sentiment.
- Determining the factors that sway whether a business is successful or not.
- Determine what users look for in businesses.

Assumptions and Limitations



Insights and Findings

- Sentiments are more obvious in the 'tips' text as opposed to the 'reviews' text.
- Certain attributes are mentioned repeatedly in reviews.
- Most users average ratings are between 3.5 - 4.5 stars.
- Most businesses average ratings are between 3.5 - 4.5 stars.

```
cvec = CountVectorizer(stop_words='english',
                       ngram_range=(1, 1))
cvec.fit(liked_tips['text'])
len(cvec.get_feature_names())
```

517

```
word_counts = tokens.sum(axis=0)
word_counts.sort_values(ascending = False).head(30)
```

like	11
place	8
just	7
free	6
food	6
refund	5
white	4
feel	4
try	4
don	4
order	4
did	4
rice	4
sucks	3
fiery	3
wait	3
sweet	3
time	3
care	3

```
vectorizer = CountVectorizer(max_features = 1000,
                             ngram_range=(3, 3),
                             stop_words='english',
                             binary=True)
vectorizer.fit(most_reviews['text'])
len(vectorizer.get_feature_names())
```

1000

```
word_counts = tokens.sum(axis=0)
word_counts.sort_values(ascending = False).head(45)
```

mon ami gabi	883
french onion soup	412
view bellagio fountains	292
filet mignon merlot	111
bloody mary bar	108
view bellagio fountain	107
great view bellagio	105
view bellagio water	76
food great service	73
corned beef hash	72
seated right away	66
steak au poivre	64
baked goat cheese	63
mignon merlot butter	59
las vegas strip	58
great food great	57
great people watching	53
service great food	51
great service great	49


```
from textblob.classifiers import NaiveBayesClassifier
```

```
train=[('great service' , 'pos'),  
       ('good food', 'pos'),  
       ('Excellent food, great atmosphere', 'pos'),  
       ('The open kitchen has been known to cause mouth watering experiences \\  
       due to the glorious aromas in an already pleasant environment', 'pos'),  
       ('We had excellent service..polite, prompt, helpful (great at helping us\  
       pick a great wine suited to our selections!) and personal.', 'pos')  
       , ('Try the little blueberry muffins and put the honey butter on \  
       anything and it tastes good!', 'pos'),  
       ('Best authentic Chinese food in all of Henderson.', 'pos'),  
       ('If you have any self respect, you should stay away from this \  
       restaurant and this horrible man.', 'neg'),  
       ('Don\'t waste your time.', 'neg'),  
       ('The food is mediocre at best but their attitude sucks', 'neg'),  
       ('The food was dry and tasteless', 'neg'),  
       ('long wait time and bad service', 'neg'),  
       ('Third Section going towards the rear of the cafe is not an ideal area to be seated.','neg')  
]  
  
cl = NaiveBayesClassifier(train)
```

```
prob_dist = cl.prob_classify(  
    "The waiter was great and definitely made the experience better.")  
round(prob_dist.prob("pos"), 2)
```

0.66

```
prob_dist = cl.prob_classify(  
    "Some of the waiters are not very friendly and the food sucks")  
round(prob_dist.prob("neg"), 2)
```

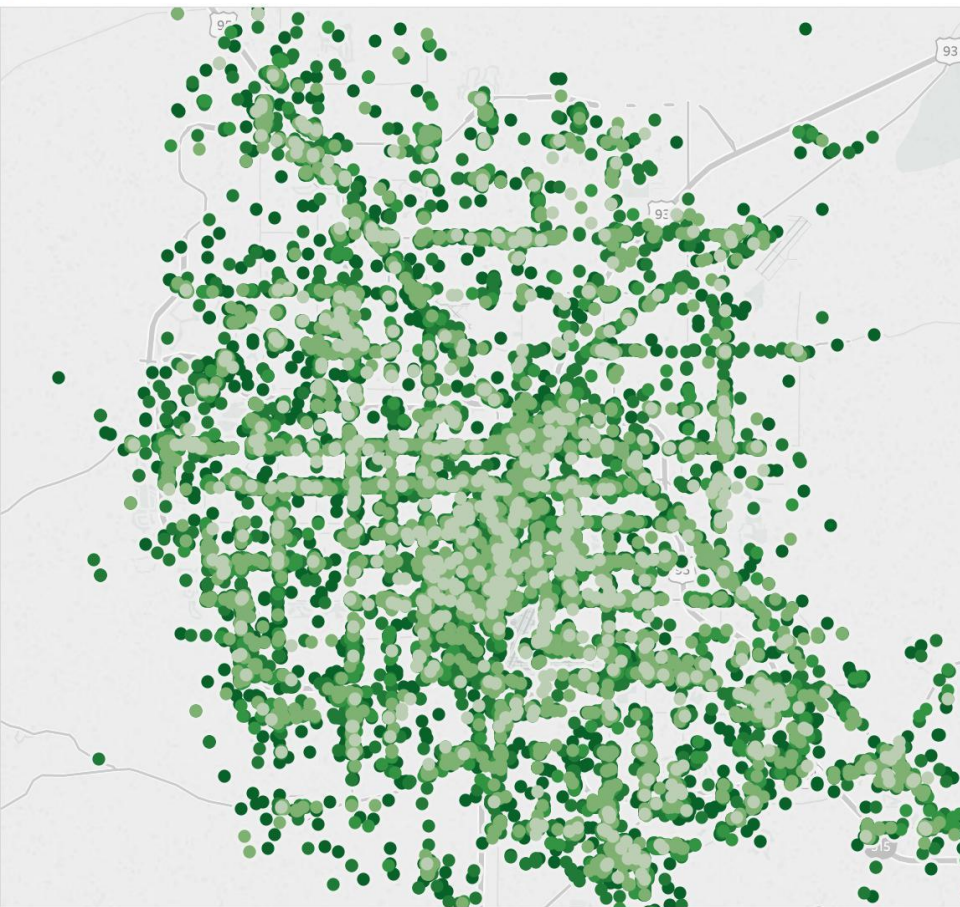
0.88

```
prob_dist = cl.prob_classify(  
    "The food here is wonderful as well as the service.")  
round(prob_dist.prob("neg"), 2)
```

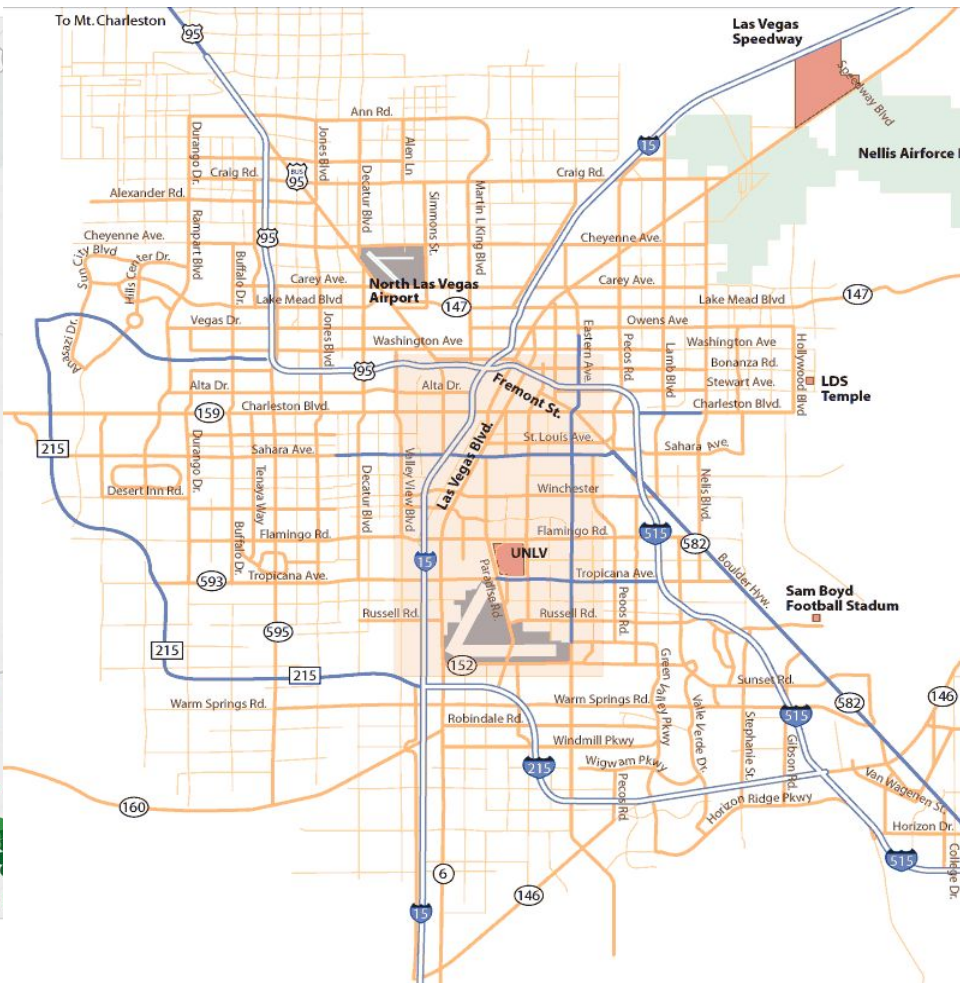
0.8

```
prob_dist = cl.prob_classify(  
    "The food here is wonderful as well as the service.")  
round(prob_dist.prob("pos"), 2)
```

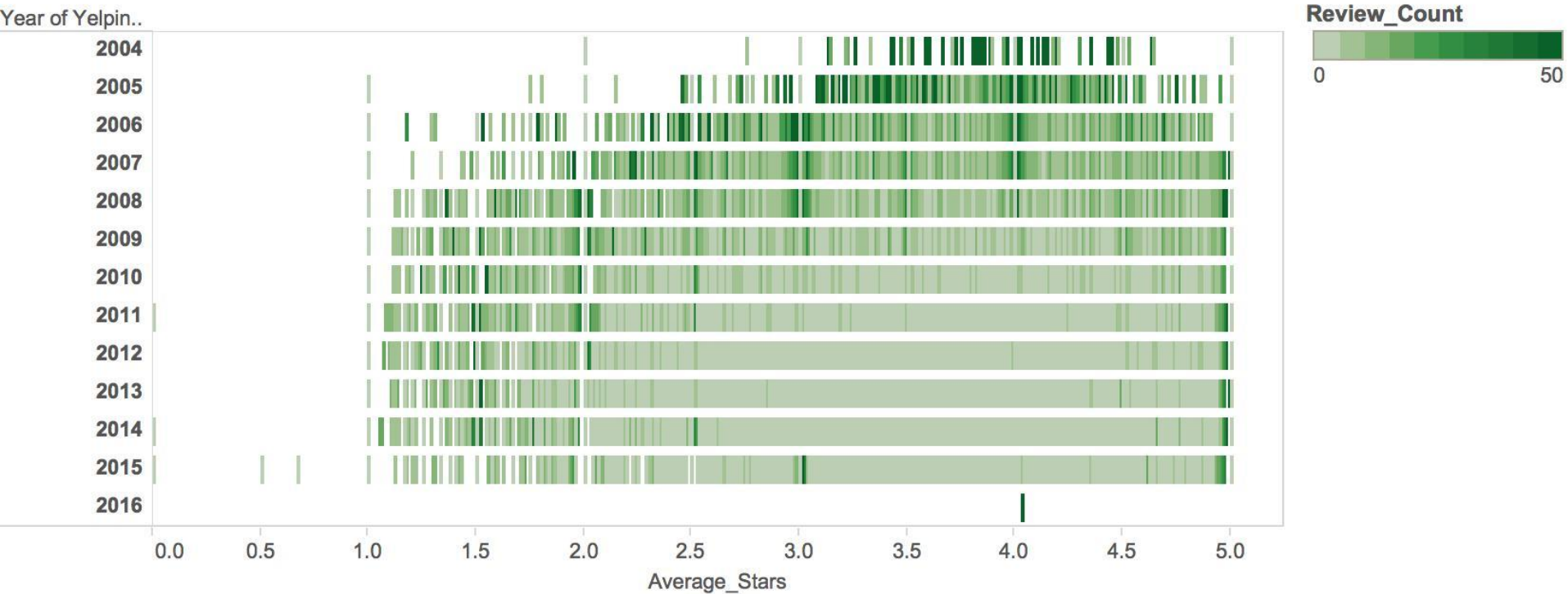
0.2



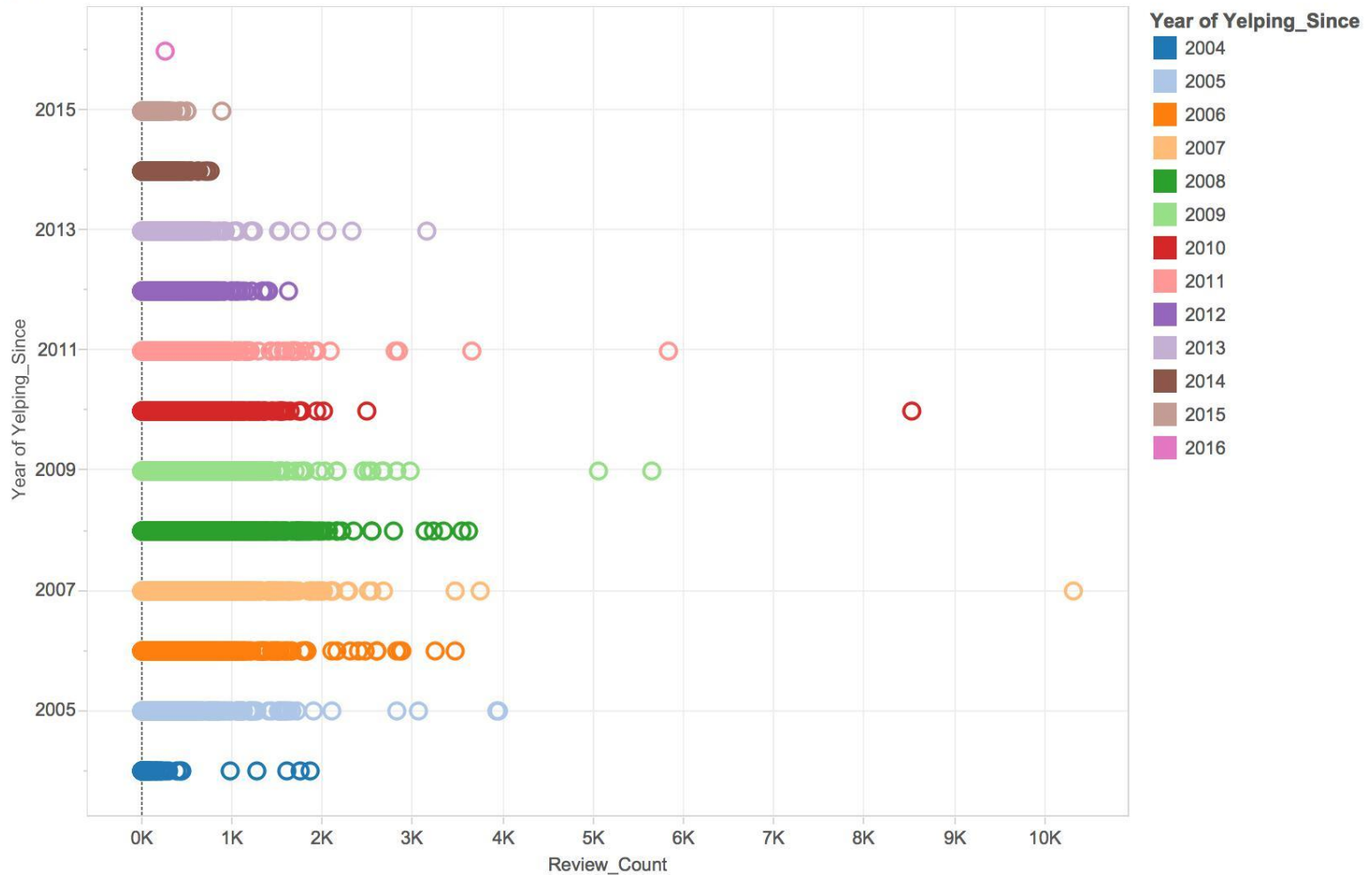
Map based on Longitude and Latitude. Color shows details about Stars.



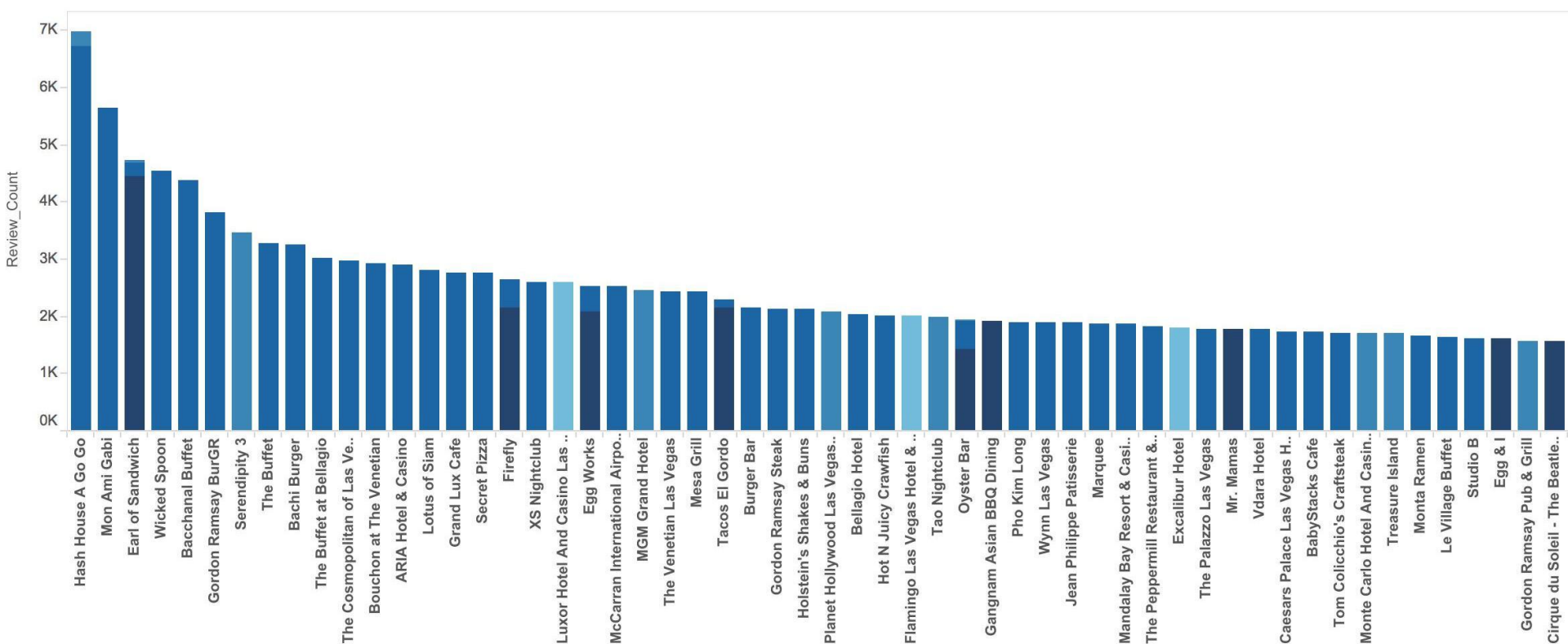
Sheet 3



Average_Stars for each Yelping_Since Year. Color shows details about Review_Count. Details are shown for Review_Count.



Review_Count vs. Yelping_Since Year. Color shows details about Yelping_Since Year.



Sum of Review_Count for each Business_Name. Color shows details about Stars. Details are shown for Stars. The view is filtered on Business_Name, which has multiple members selected.



Further Evaluation

- Check-in times, attributes, and business hours affect on business
- Elite status on credibility of the reviews and recommendations
- Compare regions.
- Incorporate computer vision.
- Make a web app using the Yelp API.