Using NLP to find the most predictive words of hotel reviews:

Define the Problem

Many reviews are based off people's opinions, can be biased, exaggerated, and highly overwhelming to constantly sift through to figure out which hotels are good and bad and what factors do people like and dislike about them. When you look at a hotel or restaurant or somewhere you want to go, you typically see a rating system giving it an average score, but what if the review text itself doesn't match the rating given by the user? This can decrease sales and reputation and also misguide new consumers in search for that new hotel or restaurant of choice. This project is aimed at building a Machine learning model using Natural Language Processing and Sentiment Analysis to predict ratings based on hotel reviews. Reviews can also help hotel hoteliers, online travel agencies, booking sites, metasearch and travel review platforms appeal more to their customer base by understanding the pros and cons of customer opinions.

Description of Data

This is a list of 1,000 hotels and their reviews provided by Datafiniti's Business Database. The dataset includes hotel location, name, rating, review data, title, username, and more.
Note that this is a sample of a large dataset. The full dataset is available through Datafiniti.
For this project, I will train and test models to predict hotel ratings and sentiment based off the users reviews that were left after their stay.

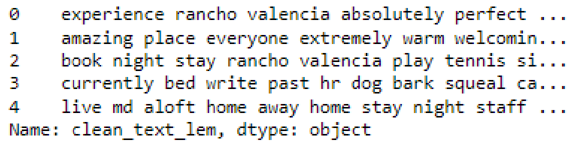
- Number of hotels: 1736
- Number of reviews: 10,000
- Rating Scales: 1 - 5, with some in decimals.

Data Wrangling

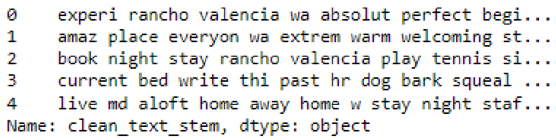
There were a few problems I ran into when cleaning the dataset. Each problem is listed below with a solution I came up with.

Problem: The text reviews have a bunch of miscellaneous information in them. This includes spelling errors, punctuation, stop words, contractions, capitalization and whitespace. Also, having different tenses of a word is not necessary. For example the words plays, played, playing all come from the base word play. I only want the base word.

Solution: To clean the text, I created a bunch of functions to lowercase the text, expand contractions, remove whitespace and remove stop words. In addition I used common practices in NLP, stemming and lemmatization to derive the base of each word. Although lemmatization took more time, I found it more effective and chose to keep the lemmatized text reviews over the stemmed ones.



```
0    experience rancho valencia absolutely perfect ...
1    amazing place everyone extremely warm welcomin...
2    book night stay rancho valencia play tennis si...
3    currently bed write past hr dog bark squeal ca...
4    live md aloft home away home stay night staff ...
Name: clean_text_lem, dtype: object
```



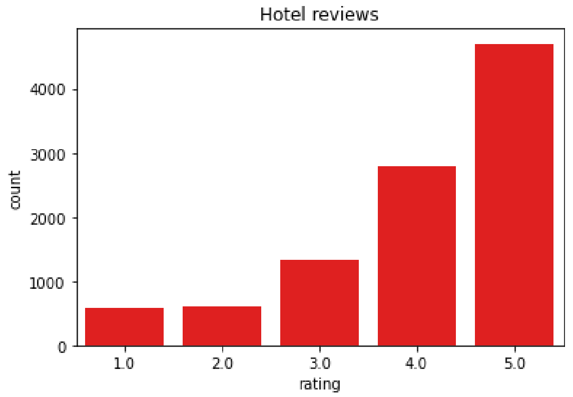
```
0    experi rancho valencia wa absolut perfect begi...
1    amaz place everyon wa extrem warm welcoming st...
2    book night stay rancho valencia play tennis si...
3    current bed write thi past hr dog bark squeal ...
4    live md aloft home away home w stay night staf...
Name: clean_text_stem, dtype: object
```

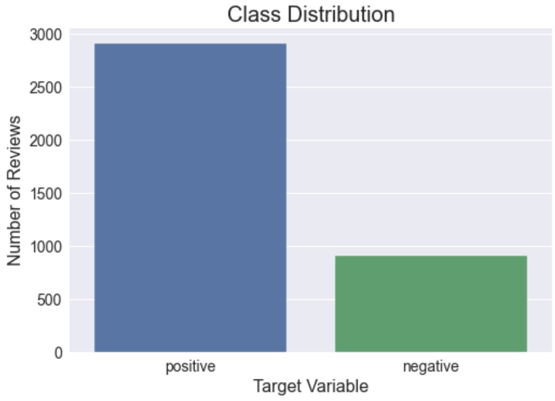
To wrap up data wrangling I did the following to the dataset:

- Rounded review ratings up and down to the appropriate whole number.
- Dropped unnecessary columns and kept review text, rating, and hotel name.
- Fixed spelling errors
- Broke down text reviews in preparation for sentiment analysis
- Added a new feature: word_count & word_count_title
- Extracted five most reviewed hotels to examine along with the entire dataset: Metro Points Hotel-Washington North ,The Westin Las Vegas Hotel & Spa, Best Western Springfield , ARIA Resort Casino, Kinzie Hotel
- Used Lang Detect to filter only english reviews.
- Started with a dataset size of 10,000 rows and 25 columns. After cleaning and removing data, it was reduced to 9998 rows and 3 columns.

More can be seen within the Data Wrangling code on my Github

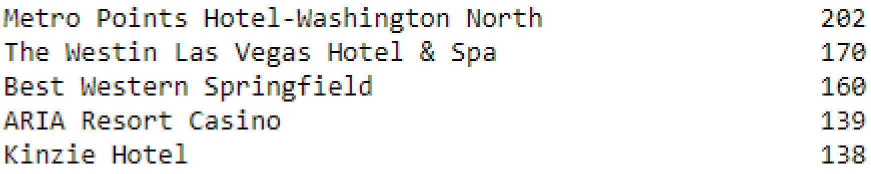
Exploratory Data Analysis



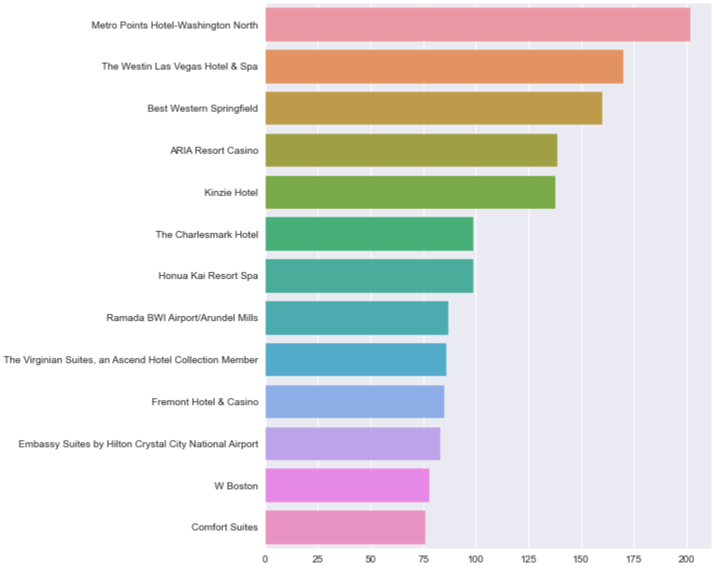


Examining the hotels I noticed that there are 1670 unique hotels within this dataset. However a majority of them do not have lots of reviews. To tackle this issue I did two things.

1) Remove any hotel with less than 25 reviews. This shrunk the number of hotels from 1670 to 76 as can be seen below. The number of reviews is 3,976 with the majority of reviews being rating 4 and 5. The top 5 hotels with the most review are:

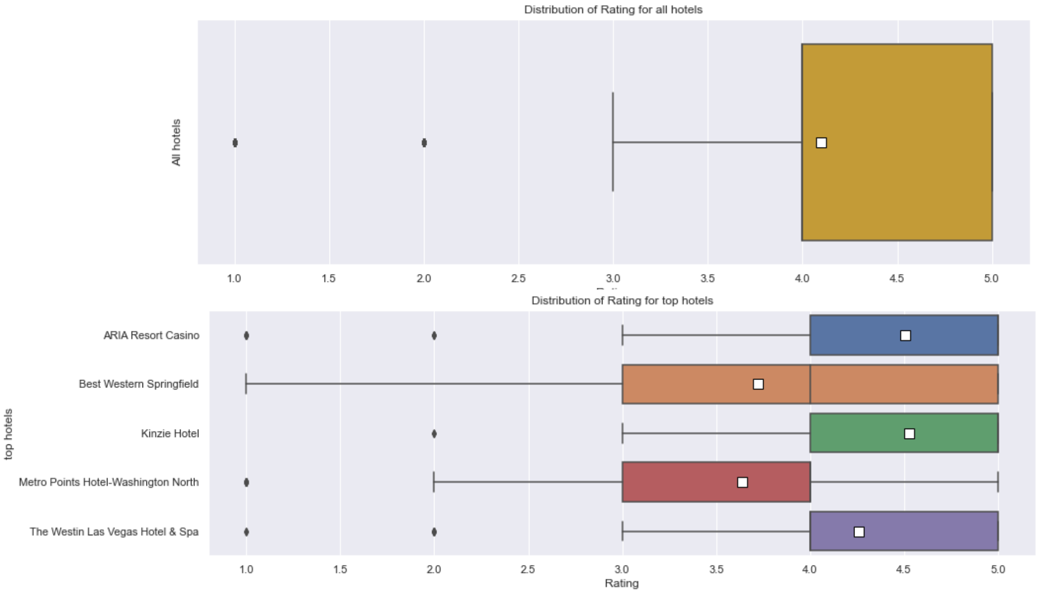


```
Metro Points Hotel-Washington North                        202
The Westin Las Vegas Hotel & Spa                           170
Best Western Springfield                                   160
ARIA Resort Casino                                         139
Kinzie Hotel                                               138
```



2) After narrowing down the most common hotels reviewed. I also chose to examine the specific words used in the top 5 hotels "Metro Points Hotel-Washington North", The Westin Las Vegas Hotel & Spa, Best Western Springfield, ARIA Resort Casino, Kinzie Hotel.

(Fig 2)



(Fig 3)

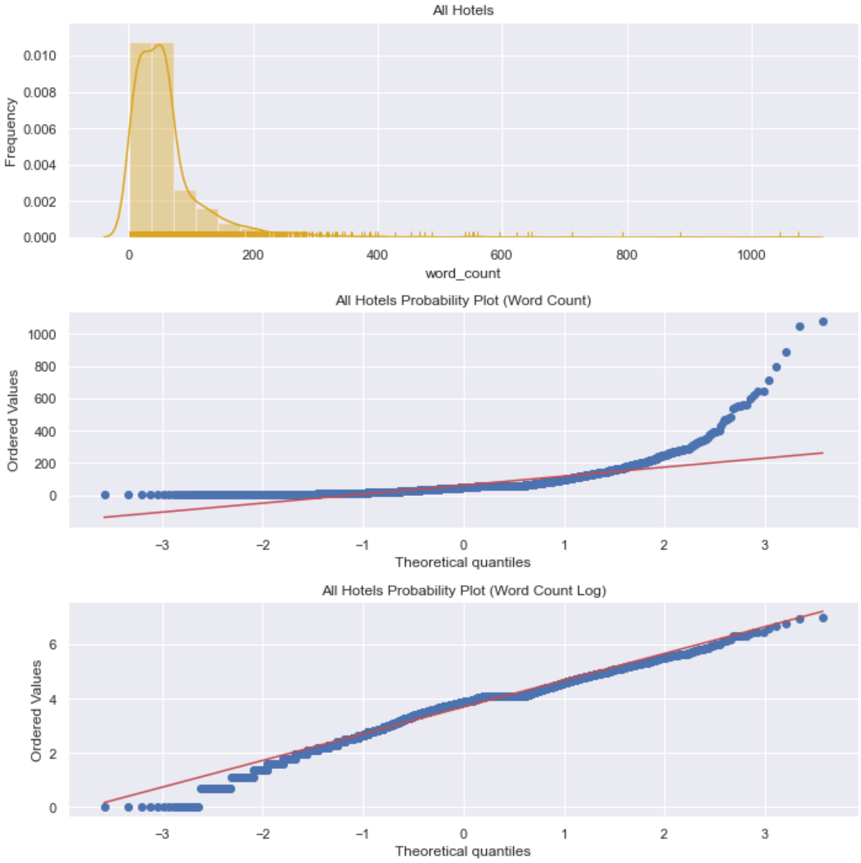
From figure 3, we can confirm our findings that for all hotels the majority of reviews are ratings between 4 and 5. Aria Resort, Kinzie Hotel, and Westin have a higher rating on average than most hotels. Best Western and Metro Points Hotel have lower ratings on average. To deal with the imbalance, I decided to group the reviews together so that reviews 4-5 are positive, and reviews 1-3 are negative. We will see how this can play into our modeling later.

Pre-Processing and Training

Engineered Features

Word Count

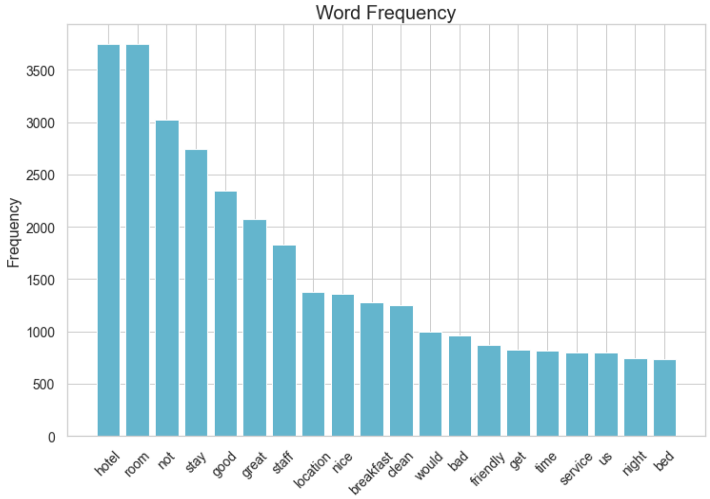
The first feature I made was the length of each text review by determining the word count. Naturally this data was skewed to the right as there were some extremely excited or utterly disappointed users who had to express their opinions in lengthy reviews. After a log-transform, the word count takes on a more normal distribution. We can see from (Fig. 4), the majority of words in reviews are less than 200 words. Out of 74 hotels and 3974 reviews. Most reviews have 25-50 words. With an average Word Count per Review of 66. Average Reviews per Hotel 54.



(Fig 4)

Word Frequencies

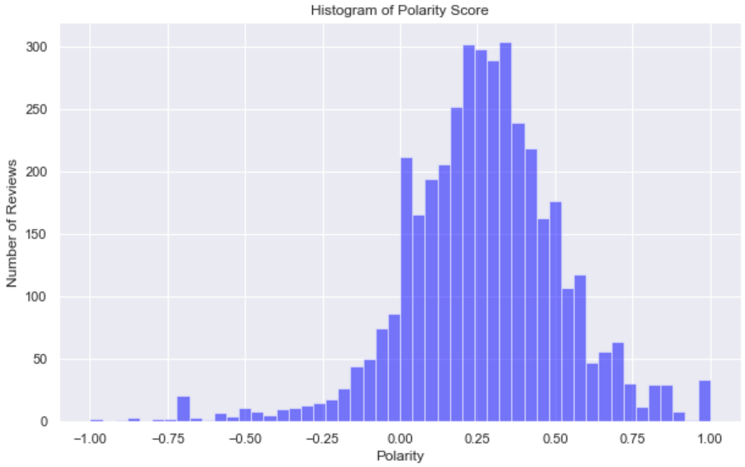
In Fig 5. We can see the frequency distribution of the 20 most abundant words in the preprocessed text corpora. Words like hotel room, great staff, nice breakfast, and others are good features of a hotel. We can also see some negative words as well as some noise.



(Fig 5)

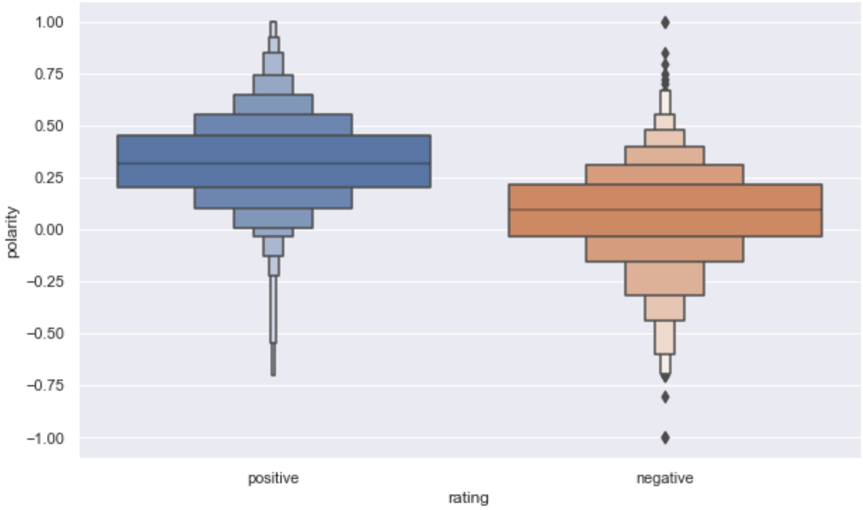
Reviews Polarity

Polarity is a float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. It is derived from the TextBlob module. Figure 3 shows the distribution of polarity score in reviews. Most of the reviews are on the positive side of the plot (Fig. 3).



(Fig. 3).

Diving deeper into reviews. Due to the imbalance of 4 and 5 ratings compared to the rest I decided to create sentiment groups with rating 1 through 3 being negative, and 4 and 5 being positive.



(Fig. 4).

In Figure 4, it can be observed that positive reviews (4s and 5s) have higher polarity compared to bad reviews. On the other hand, good reviews also have a higher number of negative and.

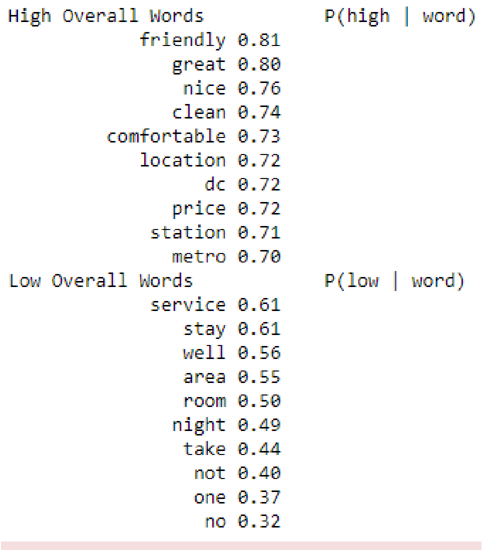
This is an unbalanced data and the number of good reviews are higher than bad reviews. Therefore, it is not much surprising to see a greater number of extreme values in this category

For Machine learning I focused on finding the most predictive words on this smaller subset of data to then apply to larger datasets as a whole. To further explore this I looked into Word Frequencies, Word Clouds and predicted probabilities of the words themselves.

Predictive Features

The most important feature of this dataset will be the text within each text review. To further understand this we will dive into each review category.

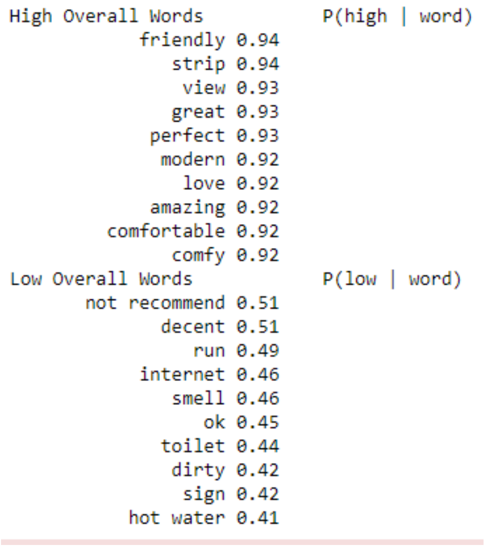
Reviews - Top Hotel - Metro Points Hotel-Washington North



```
High Overall Words          P(high | word)
            friendly 0.81
               great 0.80
                nice 0.76
               clean 0.74
         comfortable 0.73
            location 0.72
                  dc 0.72
               price 0.72
             station 0.71
               metro 0.70
Low Overall Words           P(low | word)
             service 0.61
                stay 0.61
                well 0.56
                area 0.55
                room 0.50
               night 0.49
                take 0.44
                 not 0.40
                 one 0.37
                  no 0.32
```

Let’s start first by looking at the top rated hotel. Here we can see descriptive words such as friendly, great, clean, nice. Consumers also liked features such as price, location, comfortability and even the metro is good indicators for positive experience at the hotel. Words like service, room, area, stay tell us that there can be something deeper that is affecting their stay such as maybe the service was not good or the rooms were not nice. This can give us an idea of what to look for.

Reviews - Top 5 Hotels

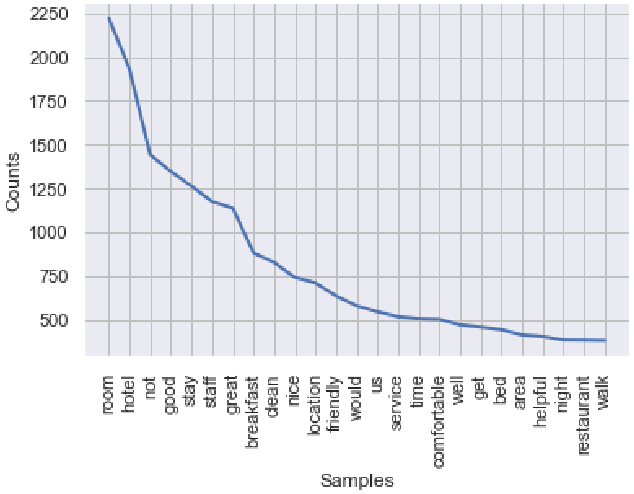


```
High Overall Words          P(high | word)
            friendly 0.94
               strip 0.94
                view 0.93
               great 0.93
             perfect 0.93
              modern 0.92
                love 0.92
             amazing 0.92
         comfortable 0.92
               comfy 0.92
Low Overall Words           P(low | word)
       not recommend 0.51
              decent 0.51
                 run 0.49
            internet 0.46
               smell 0.46
                  ok 0.45
              toilet 0.44
               dirty 0.42
                sign 0.42
           hot water 0.41
```

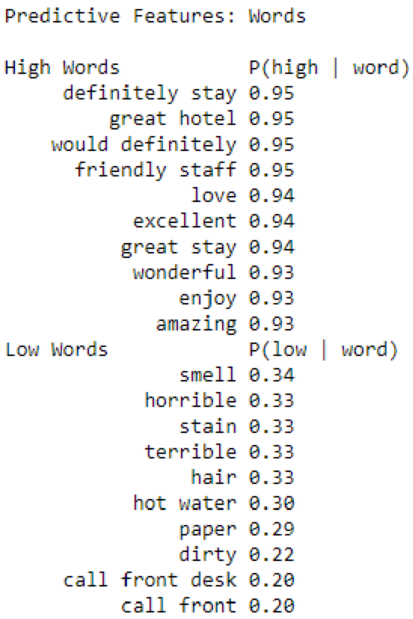
Next let’s take a look at the top 5 reviewed hotels to see if we can determine the most predictive features. We have similar descriptive words but also some specific words such as strip that are specific to the Aria hotel. We can also see some negative words such as not recommend, internet, smell, toilet, dirty. We could filter these words by review to see what is really going on if needed.

Positive Reviews - All Hotels

As mentioned earlier, the main features of the model will come from the text reviews. To further examine this I looked at the 25 most common words within the entire corpus. Among this list there are Words such as descriptive words such as good, friendly, nice, clean, helpful.

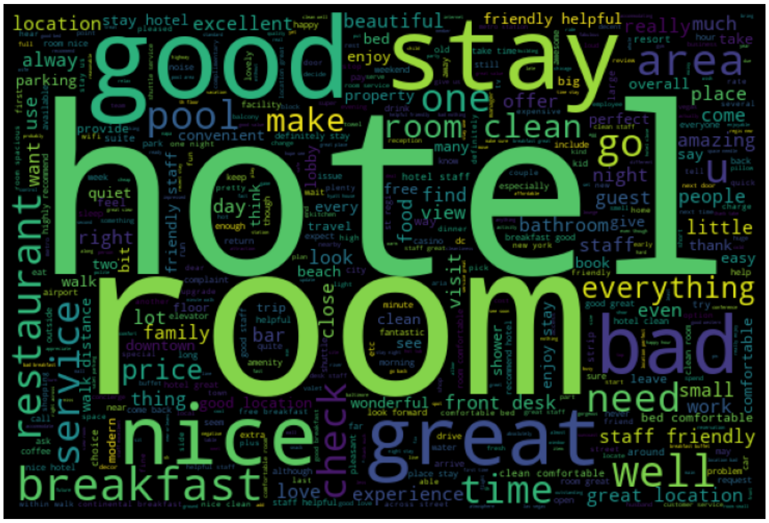


The top words are room, hotel, not, good, and stay. We can also point out positive features of a hotel which can be useful for future modeling. Positive features include rooms, food, location, service, cleanliness, and comfortability. We do also see some noise in there such as get and would and walk that we can still remove for future modeling.



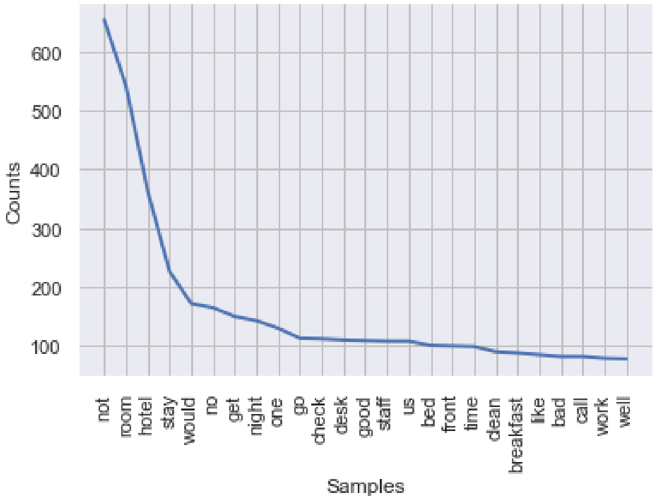
```
Predictive Features: Words

High Words          P(high | word)
     definitely stay 0.95
         great hotel 0.95
    would definitely 0.95
      friendly staff 0.95
                love 0.94
           excellent 0.94
          great stay 0.94
           wonderful 0.93
               enjoy 0.93
             amazing 0.93
Low Words           P(low | word)
               smell 0.34
            horrible 0.33
               stain 0.33
            terrible 0.33
                hair 0.33
           hot water 0.30
               paper 0.29
               dirty 0.22
     call front desk 0.20
          call front 0.20
```

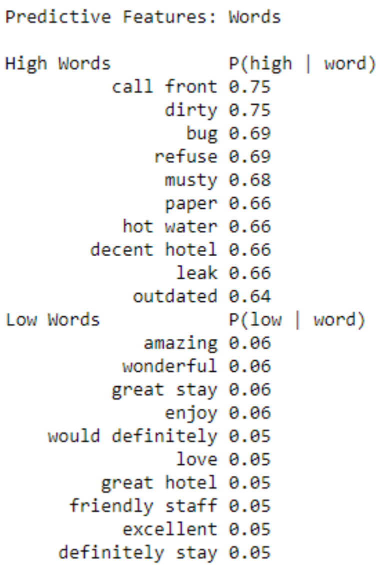


One key point to watch out for when creating a Naive Bayes model will be that the word 'not' appeared quite a bit. This may change the polarity of a review so the n-grams may play an important factor in predicting rating scales. To demonstrate this I have below the probability for words associated with a high and low overall rating. Individual words are in the first set of probabilities. You see words that you would normally see with high and low reviews such as love and good great, but you also see noise such as go front. The probabilities of each word are also very strong.

Negative Reviews - All Hotels

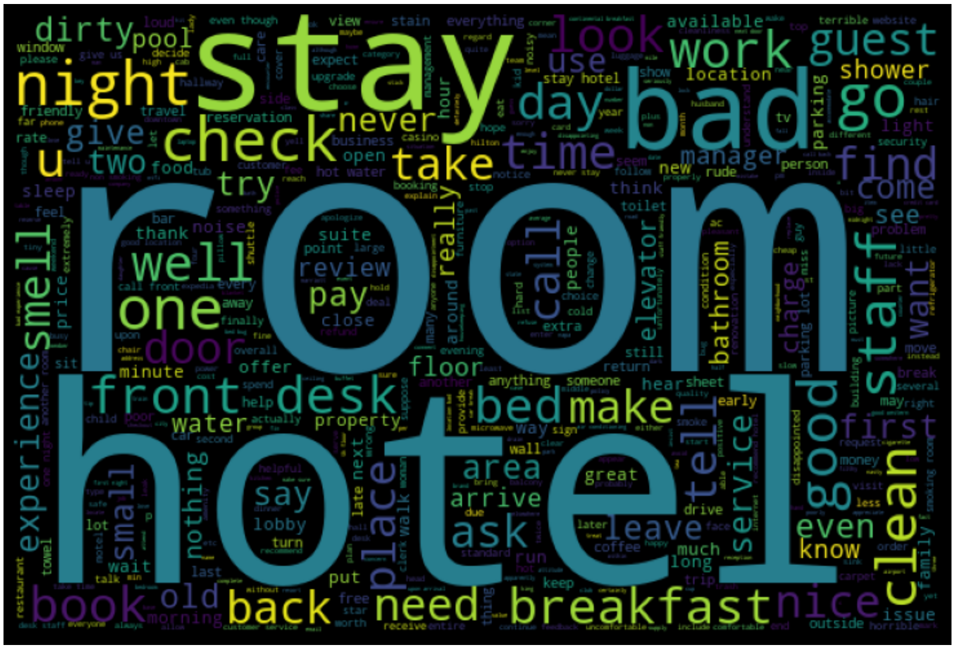


The top words are similar but have more negative connotations to them. Words like dirty, bug, musty, hot water, leak, can be descriptive in telling us what is wrong from a consumer standpoint. In comparison when you look at both one and two word phrases, you begin to get stronger probabilities. More will need to be looked when creating the models. I've added word clouds for better visualization.



```
Predictive Features: Words

High Words           P(high | word)
            call front 0.75
                 dirty 0.75
                   bug 0.69
                refuse 0.69
                 musty 0.68
                 paper 0.66
             hot water 0.66
          decent hotel 0.66
                  leak 0.66
              outdated 0.64
Low Words            P(low | word)
               amazing 0.06
             wonderful 0.06
            great stay 0.06
                 enjoy 0.06
      would definitely 0.05
                  love 0.05
           great hotel 0.05
        friendly staff 0.05
             excellent 0.05
       definitely stay 0.05
```



Machine Learning

The machine learning model aims to take new text reviews and predict either a low or high overall rating. Normally you'd first want to do some text preprocessing to clean the text reviews of nonessential information such as contractions, punctuation and whitespace, but I had already done this when first cleaning the dataset. The following steps were then taken.

Vectorizer Selection:

Vectorizers encode text data and depending on which method you choose, it is calculated differently. I decided to use two different ones, CountVectorizer and TfidfVectorizer, which are both part of Scikit-learn.

1. **CountVectorizer:** uses a "bag-of-words" approach where the text is analyzed by word counts. It counts the occurrence of each token for each review.
2. **TfidfVectorizer:** uses the term frequency(tf) and the inverse document frequency(idf) to create weighted term frequencies.

To compare the two vectorizers I applied both vectorizers separately to a simple Naive Bayes classifier, MultinomialNB() including bigrams (ngram_range = (1,3)). I calculated the ROC-AUC score to compare the models.

After I changed the minimum document frequency to .001 instead of 1. I then used GridSearchCV to select the best parameters for the simple Naive Bayes model. This told me that the default parameters for alpha and fit_prior were the best. I then recalculated the ROC-AUC score for both vectorizers.

Vectorizer	ROC-AUC	Best Parameters
CountVectorizer	0.883	min_df=1, alpha=1, fit_prior=True
TfidfVectorizer	0.835	min_df=1, alpha=1, fit_prior=True
CountVec w/ GridSearch	0.9101	min_df=.001, alpha=1, fit_prior=True
TfidfVec w/ GridSearch	0.9144	min_df=.001, alpha=1, fit_prior=True

Table 1: Comparison of vectorizers with a Multinomial Naive Bayes Model with and without hyperparameter tuning

Looking at the table above you can see that the TfidfVectorizer worked best with tuned parameters. The following parameters were best and what I used moving forward on other models and classifiers.

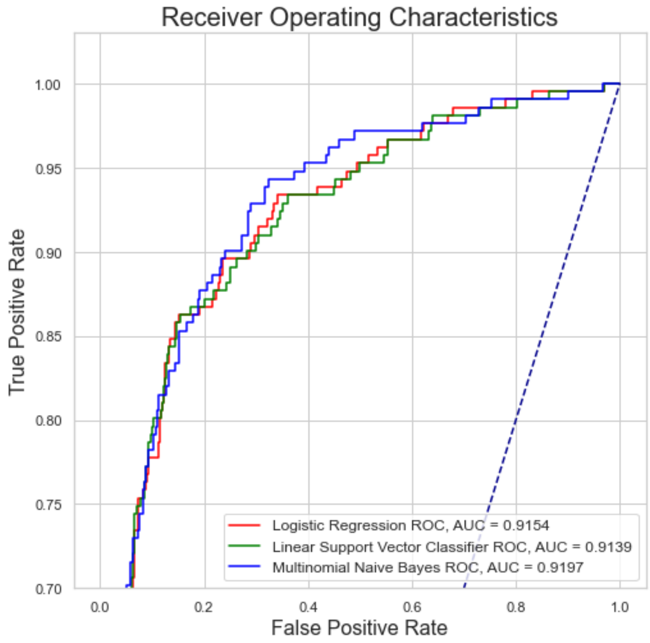
GridSearchCV(cv=5, error_score='raise-deprecating', estimator=MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True), fit_params=None, iid='warn', n_jobs=-1, param_grid={'fit_prior': (True, False), 'alpha': (0.001, 0.01, 0.1, 1, 5, 10)}, pre_dispatch='2*n_jobs', refit=True, return_train_score='warn', scoring='roc_auc', verbose=0)

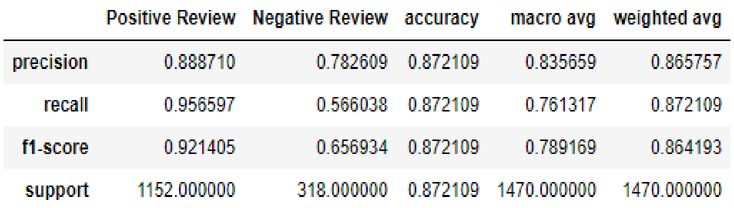
6.2. Model Comparison

Using the TfidfVectorizer I fit and tuned 3 classifiers: Logistic Regression, Support Vector Machines, Multinomial Naive Bayes. Each classifier used an ngram_range of (1,2) and a min_df of 0.001. I used the area under the curve(ROC) to compare the models to one another. I grid searched all models and chose the best one.

The highest scoring model turned out to be Naive Bayes. The score and parameters of each model are in the table below.

Classifier	ROC-AUC	Best Parameters
MultinomialNB	0.9197	alpha=0.15, fit_prior=True
LogisticRegressionCV	0.9154	C=1, max_iter=1000
Linear SVC	0.9139	C=0.1





	Positive Review	Negative Review	accuracy	macro avg	weighted avg
precision	0.888710	0.782609	0.872109	0.835659	0.865757
recall	0.956597	0.566038	0.872109	0.761317	0.872109
f1-score	0.921405	0.656934	0.872109	0.789169	0.864193
support	1152.000000	318.000000	0.872109	1470.000000	1470.000000

Conclusion

In conclusion, our capstone project on analyzing hotel reviews to predict top features of a hotel and the overall sentiment of guests has been a highly informative and enlightening experience. Through the use of natural language processing techniques, we were able to extract valuable insights from a large dataset of customer reviews, providing a deeper understanding of the key factors that influence guest satisfaction in the hospitality industry.

The analysis revealed that amenities, location, and staff service were among the most highly rated features, while cleanliness and value for money were identified as areas for improvement. These findings can be used by hotel managers to make informed decisions about how to enhance the guest experience and drive business success. By understanding the preferences and needs of their customers, hotels can implement targeted strategies to improve customer satisfaction and loyalty, leading to increased business success and profitability.

In addition to predicting the top features of a hotel, our analysis also revealed the overall sentiment of guests towards a particular hotel. By using techniques such as sentiment analysis, we were able to classify reviews as positive, or negative, providing a comprehensive overview of the overall sentiment of guests. This information can be used by hotel managers to identify areas of strength and weakness within their establishment, enabling them to make data-driven decisions to optimize their operations and improve the guest experience.

Overall, the use of natural language processing techniques has proven to be a valuable tool for predicting the top features of a hotel and the overall sentiment of guests. By leveraging the capabilities of NLP, businesses in the hospitality industry can gain a deeper understanding of their customers' preferences and needs, leading to improved customer satisfaction and loyalty. Additionally, the use of NLP can help hotels to identify patterns and trends in customer feedback, enabling them to make data-driven decisions and optimize their operations.

Furthermore, this project highlights the potential of NLP to extract valuable insights from large volumes of unstructured data. In an increasingly data-driven world, the ability to process and analyze large amounts of information is becoming increasingly important for businesses of all sizes. By using NLP to analyze customer reviews, hotels can gain a competitive advantage by better understanding the needs and preferences of their customers.

As we continue to advance the capabilities of NLP, we can unlock new and exciting opportunities for businesses to gain a deeper understanding of their customers and drive business success. By using these techniques to analyze customer feedback, hotels can make informed decisions to enhance the guest experience and drive business success. In short, the use of NLP in the hospitality industry has the potential to revolutionize the way in which hotels understand and interact with their customers, leading to improved customer satisfaction and loyalty, and ultimately, increased business success.