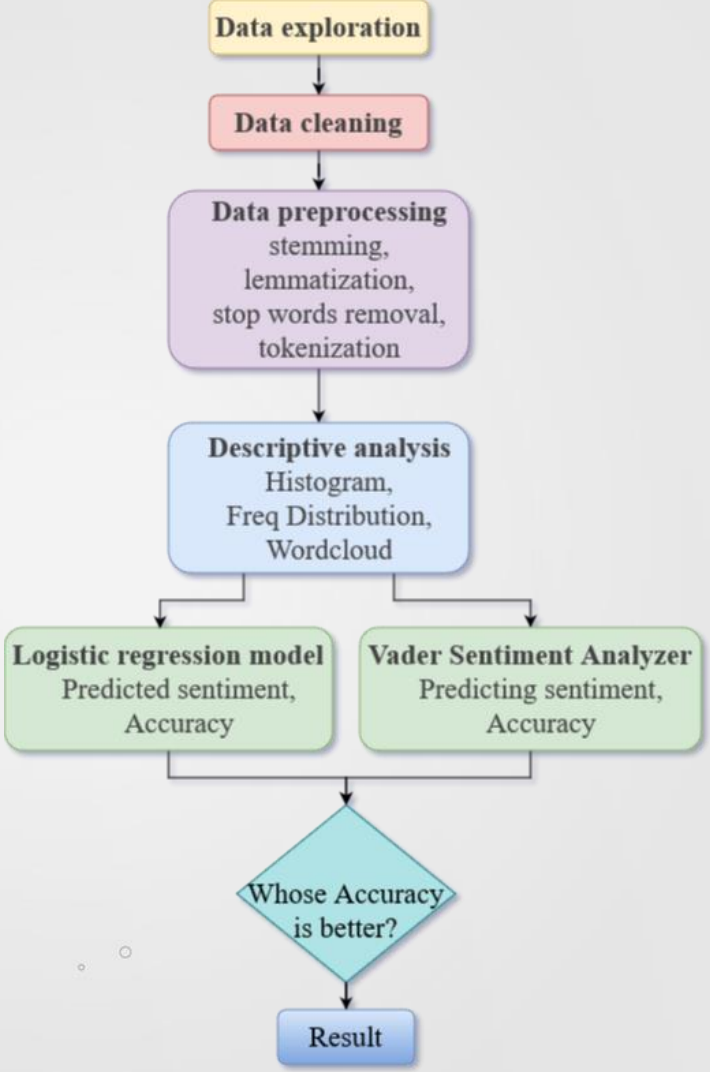
BAN 200- ZAA Sentiment Analysis and Text Mining

Rahil Ansari | Parth Shah | Rangeetha | Melissa Pinheiro | Shivani Nanavati

GOAL

Final Group Project

- Dataset used – Hotel reviews from Kaggle.
- Exploratory Data Analysis using ML to understand and extract the useful information from the reviews.
- Performing Sentiment analysis on hotel reviews, to find if it's a positive or a negative review, using two machine learning techniques - **logistic regression model** and **Vader sentiment analyzer** and determining the best model



Analytical process

Steps followed for analysis:

Data Analysis of Text Data including Visualization

Our Approach

Data Loading & Cleaning

- Checking for null values
- Removing the null values

```
hotel  = hotel[['name', 'reviews.rating', 'reviews.text']]

hotel.head()
```

	name	reviews.rating	reviews.text
0	Hotel Russo Palace	4.0	Pleasant 10 min walk along the sea front to th...
1	Hotel Russo Palace	5.0	Really lovely hotel. Stayed on the very top fl...
2	Hotel Russo Palace	5.0	Ett mycket bra hotell. Det som drog ner betyge...
3	Hotel Russo Palace	5.0	We stayed here for four nights in October. The...
4	Hotel Russo Palace	5.0	We stayed here for four nights in October. The...

Checking null values in the rows

```
print(len(hotel) - len(hotel.dropna()))
```

884

Removing the null values

```
hotel = hotel.dropna()
len(hotel)
```

Data Loading & Cleaning

- Checking the count of hotel reviews.

In [6]:
```
hotel['name'].value_counts()
```

Out[6]:
```
The Alexandrian, Autograph Collection    1185
Howard Johnson Inn - Newburgh             714
Americas Best Value Inn                   566
Fiesta Inn and Suites                     546
Ip Casino Resort Spa                      392
                                         ...
Petretti Apartments                         1
Nesco Manor Hotel                           1
Brooks Donald L Jr                          1
Days Inn Marion                             1
Regency Inn Motel                           1
Name: name, Length: 792, dtype: int64
```

Some hotels have only 1 review and hence its not possible to draw conclusion out of 1 review and hence considering the hotels with atleast reviews more than 25

In [7]:
```
hotel = hotel[hotel.groupby("name")["name"].transform('size') > 25]
```

In [8]:
```
len(hotel)
```

Out[8]:
```
32197
```

Data Preprocessing

- Deleting the special characters
- Converting all text to lower case

Deleting the Special Characters:Reviews may contain special characters which are not helpful for analysis ,hence cleaning them.

In [12]:
```
def clean(txt):
    txt = txt.str.replace("(<br/>)", "")
    txt = txt.str.replace('(<a).*(>).*(</a>)', '')
    txt = txt.str.replace('(&amp)', '')
    txt = txt.str.replace('(&gt)', '')
    txt = txt.str.replace('(&lt)', '')
    txt = txt.str.replace('(\xa0)', ' ')
    return txt
hotel['reviews.text'] = clean(hotel['reviews.text'])
```

Converting to lower case so that for eg: the and The are not considered as different words

In [13]:
```
hotel['reviews1.text'] = hotel['reviews.text'].apply(lambda x: " ".join(x.lower() for x in x.split()))
hotel['reviews1.text'].head()
```

Out[13]:
```
0    pleasant 10 min walk along the sea front to th...
1    really lovely hotel. stayed on the very top fl...
2    ett mycket bra hotell. det som drog ner betyge...
3    we stayed here for four nights in october. the...
4    we stayed here for four nights in october. the...
Name: reviews1.text, dtype: object
```

Data Preprocessing

- Removing stop words
- Stemming and lemmatization

In [15]:
```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')
hotel['reviews1.text'] = hotel['reviews1.text'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
hotel['reviews1.text'].head()
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\RAHIL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[15]:
```
0    pleasant 10 min walk along sea front water bus...
1    really lovely hotel. stayed top floor surprise...
2    ett mycket bra hotell. det som drog ner betyge...
3    stayed four nights october. hotel staff welcom...
4    stayed four nights october. hotel staff welcom...
Name: reviews1.text, dtype: object
```

All the stopwords have been removed now.

Stemming and lemmatization ,cutting down the parts like 'ly','ing' etc

In [16]:
```
from nltk.stem import PorterStemmer
st = PorterStemmer()
hotel['reviews1.text'] = hotel['reviews1.text'].apply(lambda x: " ".join([st.stem(word) for word in x.split()]))
```

In [19]:
```
import nltk
nltk.download('wordnet')
from textblob import Word
hotel['reviews1.text'] = hotel['reviews1.text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))
hotel['reviews1.text'].head()
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\RAHIL\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
```

Out[19]:
```
0    pleasant 10 min walk along sea front water bus...
1    realli love hotel. stay top floor surpris jacu...
2    ett mycket bra hotell. det som drog ner betyge...
3    stay four night october. hotel staff welcoming...
4    stay four night october. hotel staff welcoming...
Name: reviews1.text, dtype: object
```

Data Preprocessing

- Removing the punctuations

Removing the punctuations

In [21]:
```
hotel['reviews1.text'] = hotel['reviews1.text'].str.replace('[^\w\s]', '')
hotel['reviews1.text'].head()
```

Out[21]:
```
0    pleasant 10 min walk along sea front water bus...
1    realli love hotel stay top floor surpris jacuz...
2    ett mycket bra hotell det som drog ner betyget...
3    stay four night october hotel staff welcoming ...
4    stay four night october hotel staff welcoming ...
Name: reviews1.text, dtype: object
```

Data Analysis

- adding new columns like length of the review, word count and polarity of the reviews

In [22]:

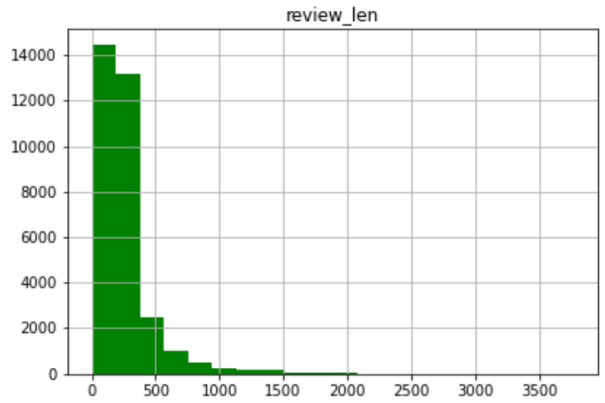
```
hotel['review_len'] = hotel['reviews.text'].astype(str).apply(len)
hotel['word_count'] = hotel['reviews.text'].apply(lambda x: len(str(x).split()))
from textblob import TextBlob, Word, Blobber
hotel['polarity'] = hotel['reviews1.text'].map(lambda text: TextBlob(text).sentiment.polarity)
hotel.head()
```

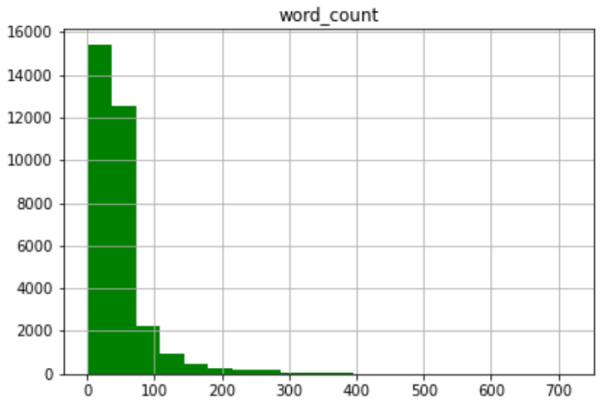
Out[22]:

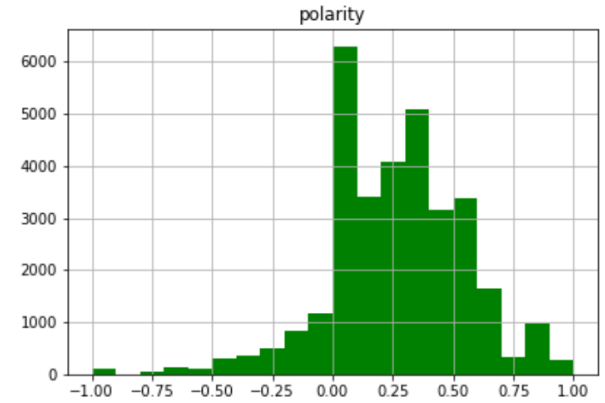
	name	reviews.rating	reviews.text	reviews1.text	review_len	word_count	polarity
0	Hotel Russo Palace	4.0	Pleasant 10 min walk along the sea front to th...	pleasant 10 min walk along sea front water bus...	194	33	0.716667
1	Hotel Russo Palace	5.0	Really lovely hotel. Stayed on the very top fl...	realli love hotel stay top floor surpris jacuz...	252	44	0.680000
2	Hotel Russo Palace	5.0	Ett mycket bra hotell. Det som drog ner betyge...	ett mycket bra hotell det som drog ner betyget...	136	28	0.350000
3	Hotel Russo Palace	5.0	We stayed here for four nights in October. The...	stay four night october hotel staff welcoming ...	354	59	0.309524
4	Hotel Russo Palace	5.0	We stayed here for four nights in October. The...	stay four night october hotel staff welcoming ...	354	59	0.309524

Data Analysis

- Distribution of length of review, word count and polarity

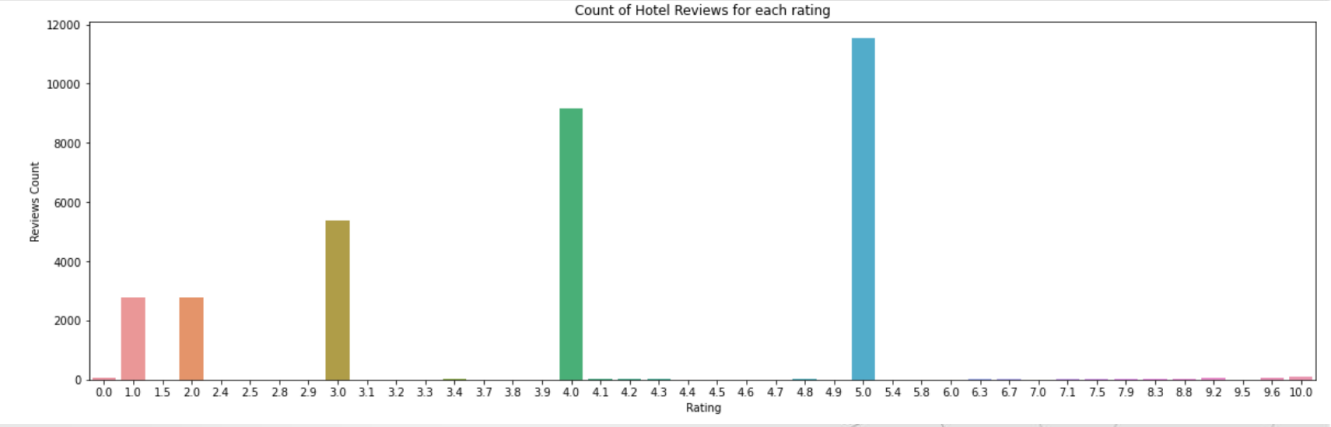






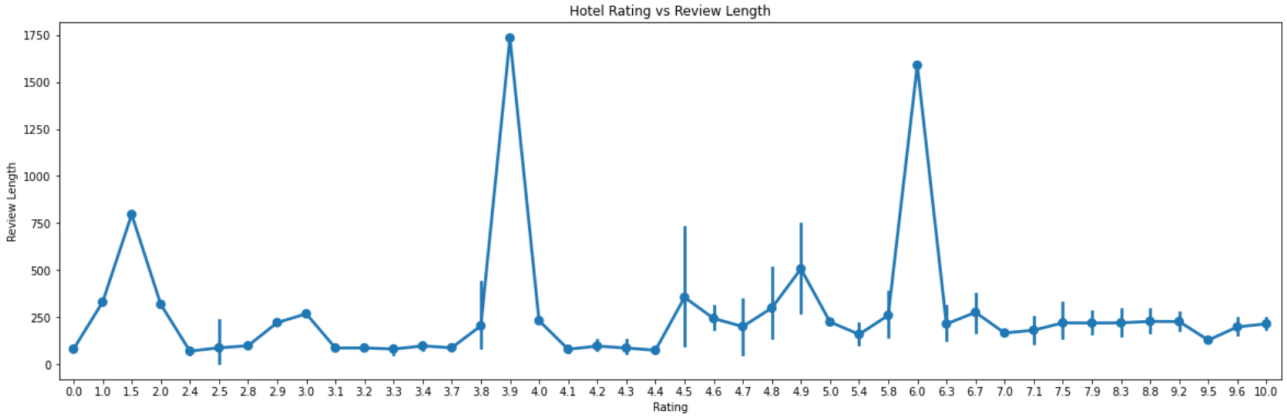
Data Analysis

- Number of reviews for each rating



Data Analysis

- Hotel ratings vs review length
- Checking if review length changes with rating



When the rating is 4 and 6, review length significantly goes up.But as the rating increaese beyond 6 ,the review length goes down. so when customers were happy, they didn't write too much!

Data Analysis

- The top 20 hotels based on polarity

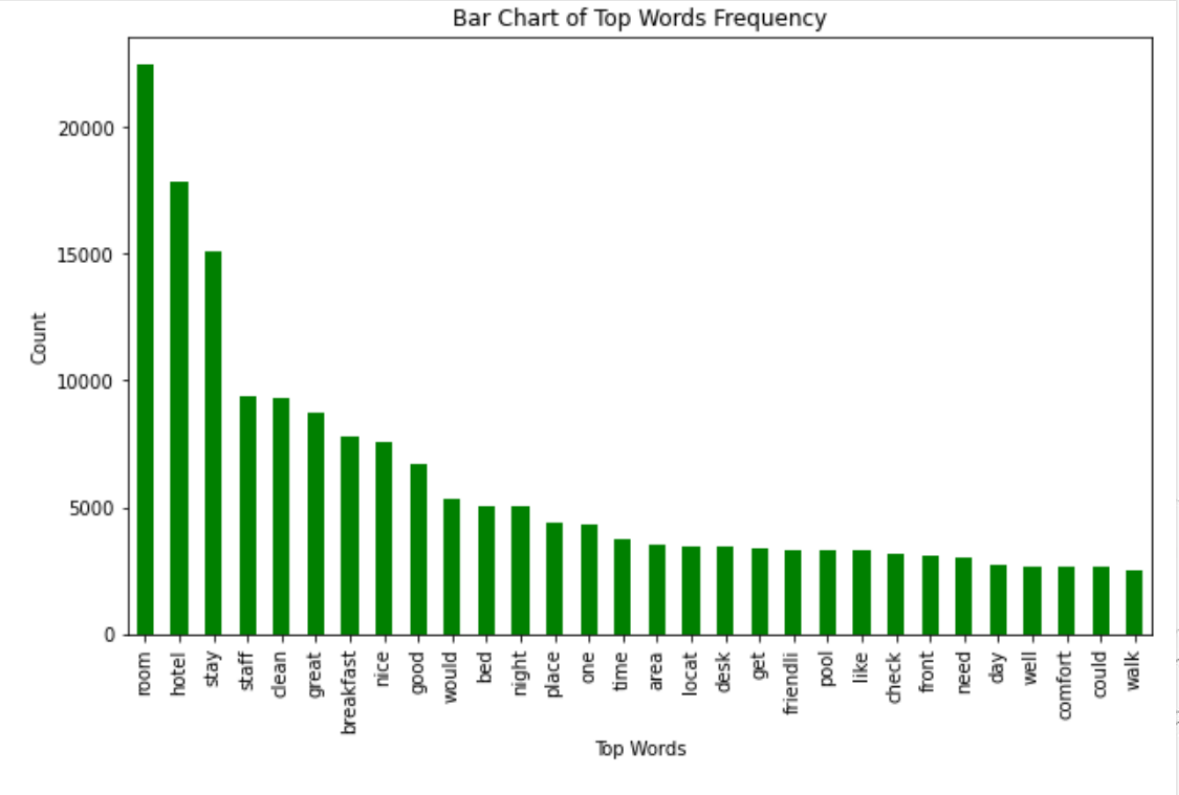
name	polarity
The Inn At Bella Vista	0.476999
Hyatt Place Pittsburgh Cranberry	0.432823
Inturotel Esmeralda Park	0.418810
Springhill Suites Marriott Colorado Springs South	0.418161
Staybridge Suites Tyler University Area	0.416990
Comfort Inn Deland - Near University	0.416103
Doubletree By Hilton Hotel Bay City - Riverfront	0.411125
Candlewood Suites Lexington	0.410749
Merritt House Inn	0.404777
Hotel Mc Call	0.399033
La Quinta Inn & Suites Bryant	0.391952
Holiday Inn Express Hotel and Suites Meadowlands Area	0.382875
The Westin Europa and Regina	0.382373
Residence Inn By Marriott Irvine John Wayne Airport	0.380465
Americinn Lodge Suites Princeton	0.378222
Hampton Inn Virginia Beach Oceanfront North	0.376358
Gran Melia Victoria	0.375526
Hampton Inn Grand Junction Downtown/historic Main Street	0.375401
Country Inn and Suites By Carlson Galena	0.372778
Hampton Inn New Orleans - Downtown	0.368990

Data Analysis

- Word cloud to check the frequency of words used in the reviews

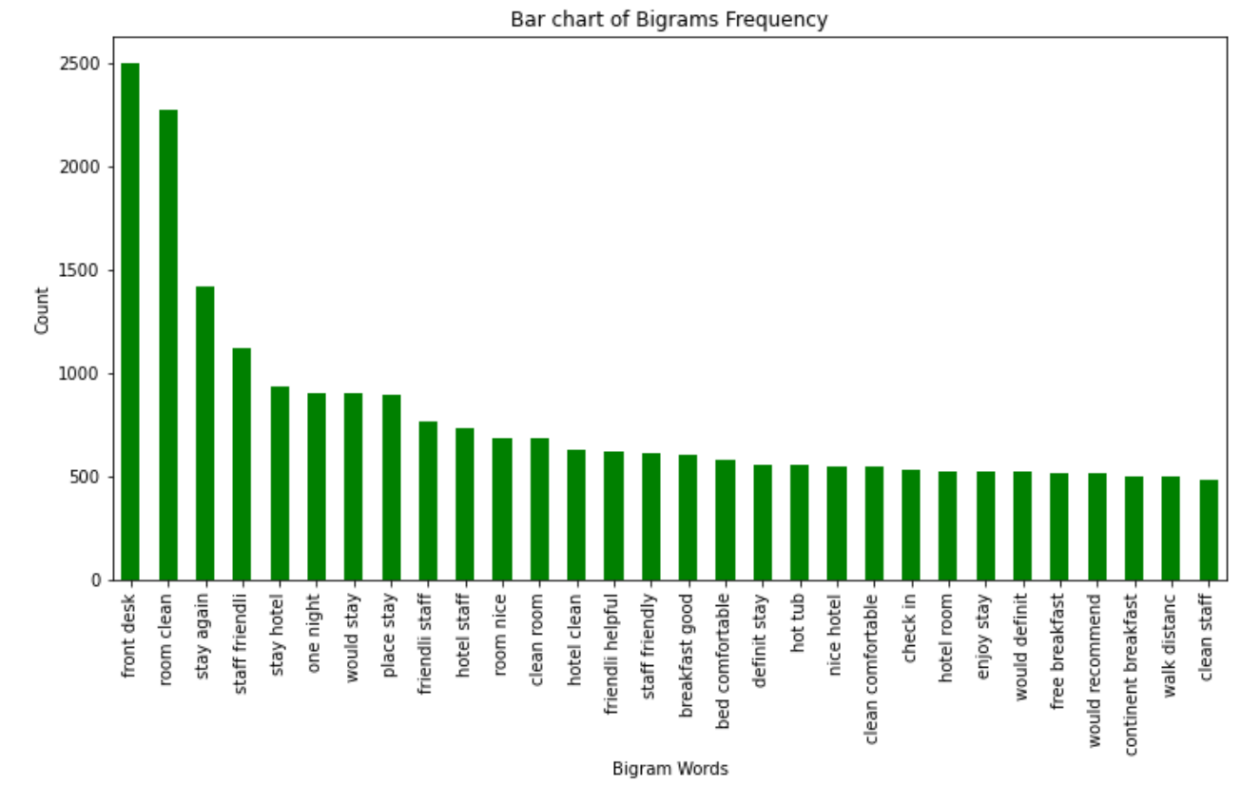
Data Analysis

- The top 30 words based on frequency



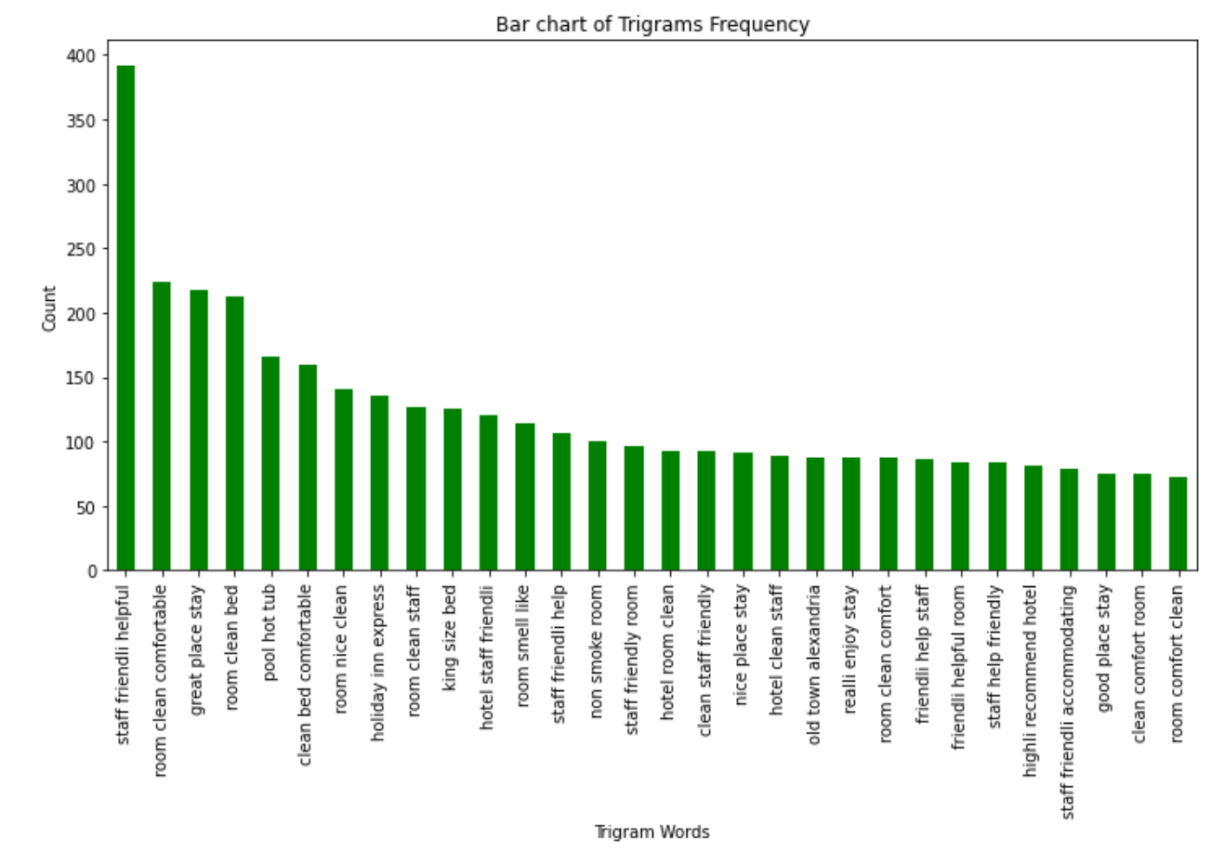
Data Analysis

- The top 30 bigram words based on frequency



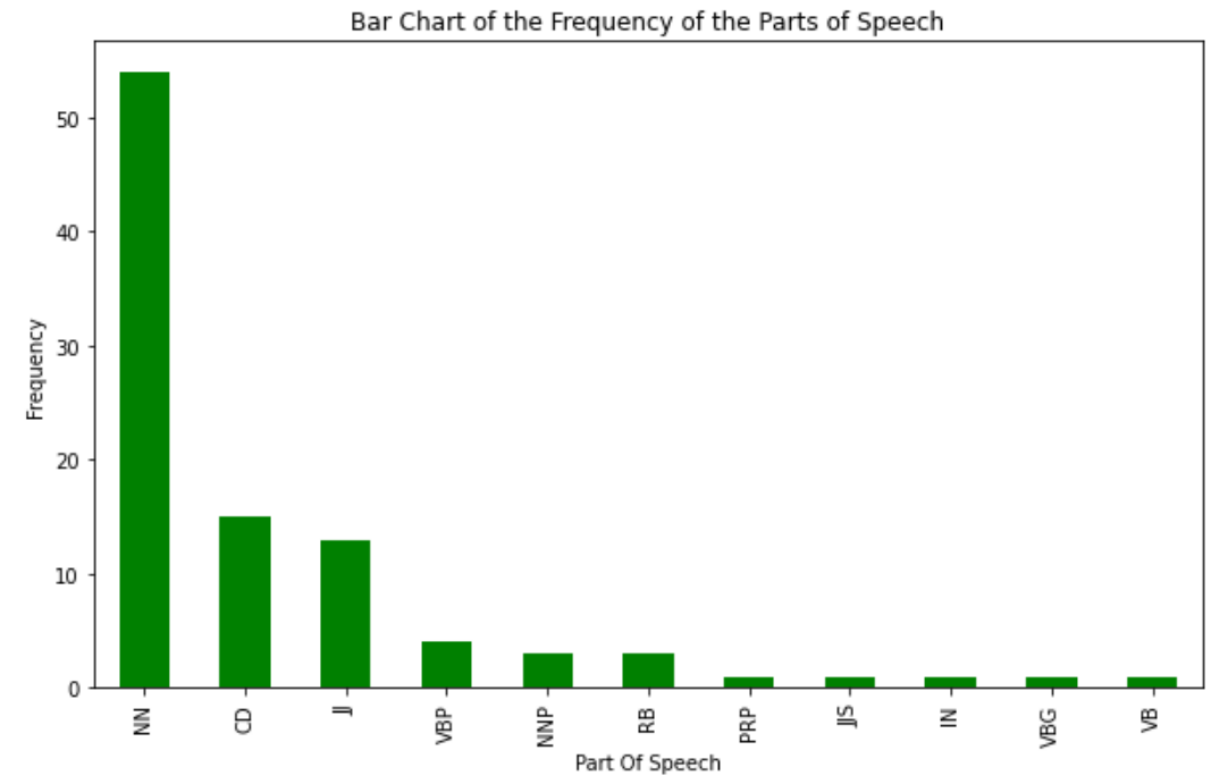
Data Analysis

- The top 30 trigram words based on frequency



Data Analysis

- Parts of speech tagging such as nouns, pronouns, verbs, adjectives etc



Data Analysis

- Reviews with rating 3 were removed to avoid ambiguity
- sentiment column created where reviews with rating 4/5 were considered positive and tagged as 1 and reviews with rating 1/2 were considered as negative and tagged as 0

Removing reviews that have rating 3 out of 5 because it can be ambigous i.e. neither positive nor negative

```
len(hotel)
```

```
[2]: 32197
```

```
hotel = hotel [hotel['reviews.rating'] !=3]
len(hotel)
```

```
[3]: 26813
```

Creating a sentiment column inside hotel dataframe to classify the type of the review.

If the hotel got 4 or 5 rating then its considered a positive review and tagged as 1 and If the hotel got 1 or 2 rating then its considered a negative review an tagged as 0

Building the sentiment classifier- Logistic regression

Import train_test_split	Build Training and testing dataset. 7:3 ratio
Import count Vectorizer	Converting review into vector
Import LogisticRegression	classifies a review as either positive or negative based on given sentiment column and text review
model = LogisticRegression(max_iter=1000)	commanding to iterate a 1000 time to ensure better accuracy.
y_predicted = model.predict(x_test_dtm) type(y_test)	predicting value of Y(test dataset)
accuracy_score(y_test, y_predicted)	The score generated is 0.914745600873005

VADER

- Import list of positive and negative words from external sources
- Perform Tokenization, stemming and text cleaning
- SentimentIntensityAnalyzer() function.
- ‘Results’ – a variable used to store scores
- results = pd.DataFrame(results.tolist())- to compound polarity
- Introducing a vader sentiment column
- Check accuracy. - 68 percent

Source: https://towardsdatascience.com/exploratory-data-analysis-of-text-data-including-visualization-and-sentiment-analysis-e46dda3dd260

OUTCOME

- Vader sentiment analyzer – 68% Accuracy
- logistic regression model which was 91.47%.
- The two rational –

a. Ambiguity
b. Noise in the dataset.

CONCLUSION

- Preprocessing techniques
- Understanding sentiments and trends – repetitive words
- Discovering and visualizing exploitable pattern- Co-relations, composition of content.
- Finding best ML model to analyze sentiments.

References

- https://towardsdatascience.com/exploratory-data-analysis-of-text-data-including-visualization-and-sentiment-analysis-e46dda3dd260
- https://www.kaggle.com/datafiniti/hotel-reviews
- https://towardsdatascience.com/a-complete-sentiment-analysis-algorithm-in-python-with-amazon-product-review-data-step-by-step-2680d2e2c23b
- https://towardsdatascience.com/design-your-own-sentiment-score-e524308cf787