MDA 720

HOTEL REVIEW SENTIMENT ANALYSIS

Zubair Hossain Mahamud

TABLE OF CONTENTS

Background:

The hospitality industry is highly competitive, and customers are looking for quality and personalized services when choosing their accommodations. In this context, hotel reviews and online feedback play a crucial role in shaping the perception of potential customers. Sentiment analysis is a technique that can be used to analyze large volumes of hotel reviews, classify them into positive, negative, or neutral categories, and identify the areas where the hotel can improve to enhance customer satisfaction. The objective of this project is to use sentiment analysis to analyze hotel reviews and provide recommendations for a new hotel business.

Objective/Goals of the Project:

The goals of this project are:

- To collect a large dataset of hotel reviews from online platforms.
- To perform sentiment analysis on the reviews and classify them into positive, negative, or neutral categories.
- To identify the areas where the hotel can improve to enhance customer satisfaction.
- To provide recommendations for a new hotel business based on the analysis.

Collecting Data:

The initial step is to load the raw data. There is a positive and a negative section to each textual review. To begin with only raw text data and no other information, we group them together. The data can be found here: https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe

	review	is_bad_review
0	I am so angry that i made this post available...	1
1	No Negative No real complaints the hotel was g...	0
2	Rooms are nice but for elderly a bit difficul...	0
3	My room was dirty and I was afraid to walk ba...	1
4	You When I booked with your company on line y...	0

After reading the dataset I narrow the dataset and read-only related columns from the dataset. This data shows a binary classification model to predict whether a given review is a bad review or not, based on the text content of the review. This can be done using various natural language processing (NLP) techniques, such as text cleaning, tokenization, feature extraction, and machine learning algorithms such as logistic regression or support vector machines. I have used the text-cleaning process in this case.

In our data, this will be marked as "No Negative" if the user does not provide any negative feedback. The positive comments have the same effect, with the default setting set to "No Positive." Those sections must be removed from our texts.

Data Cleaning:

Cleaning data is crucial because it ensures that it is reliable, consistent, and accurate. Data errors, inconsistencies, and inaccuracies, such as missing values, duplicates, outliers, and formatting issues, must be identified and corrected. Organizations can improve the quality of their data and avoid mistakes that could result in incorrect analyses and decisions by cleaning it. Additionally, biases in machine learning algorithms, which can result in unfair outcomes, can be avoided with clean data. In conclusion, data cleaning is necessary to guarantee that the data are reliable and can be used effectively for analysis and decision-making.
In our data, this will be marked as "No Negative" if the user does not provide any negative feedback. The positive comments have the same effect, with the default setting set to "No Positive." Those sections must be removed from texts.

I use our custom "clean_text" function, which carries out a number of transformations, to clean textual data:

- bring down the text
- tokenize the text (split the text into words) and eliminate the accentuation
- eliminate pointless words that contain numbers
- eliminate pointless stop words like 'the', 'a' ,'this' and so forth.
- POS (part of speech) tagging relegates a tag to each word to characterize in the event that it compares to a thing, an action word, and so forth. utilizing the WordNet lexical data set
- lemmatize the text: transform each word into its root form, such as rooms -> room or slept -> sleep. Now that our data has been cleaned, we can begin the modulization phase with some feature engineering.

Sentiment Analysis:

A feeling examination is essential for the Normal Language Handling (NLP) methods that comprise separating feelings connected with a few crude messages. This is usually used in customer reviews and posts on social media to automatically determine whether certain users are positive or negative and why. This study aims to demonstrate how sentiment analysis can be carried out with Python.

Some of the main libraries we'll use are as follows:

NLTK: Genism is the most well-known Python module for NLP techniques: a subject demonstrating and vector space displaying toolbox.

Scikit-learn: the Python machine learning library that is used the most We will use some hotel reviews data in this case. One customer review for a single hotel is included in each observation. A textual review of the customer's stay at the hotel and an overall rating is included in each customer review.

For each printed survey, we need to foresee if it relates to a decent survey or to a terrible one. The overall ratings for the reviews can be anywhere from 2.5/10 to 10/10. To simplify the issue, we will divide those into two groups:

The challenge here is to be able to predict this information using only the raw textual data from the review. Good reviews have overall ratings greater than or equal to 5.

Step 1:
Since it can be decided by inferring that customer reviews are strongly linked to how they felt about their stay at the hotel, we will first begin by adding features for sentiment analysis. Vader is the sentiment analysis component of the NLTK module that we use. Vader utilizes a dictionary of words to find which ones are up-sides or negatives. It additionally considers the setting of the sentences to decide the opinion scores. Vader returns four values to each text:

a neutrality scores
a positivity scores
a negativity scores
an overall score that summarizes the previous scores.

Those four values will be incorporated as features into our dataset.

Step 2:
After that, we include a few basic metrics for each text, the total number of characters and words in the text
Step 3:

The extraction of vector representations for each review is the next step. The module Gensim makes a mathematical vector portrayal of each and every word in the corpus by involving the settings where they show up (Word2Vec). Using shallow neural networks, this is done. Fascinating those comparative words will have comparable portrayal vectors.

Using the word "vectors" (Doc2Vec), each text can also be transformed into numerical vectors. The same messages will likewise have comparable portrayals and to that end, we can involve those vectors in preparing highlights.

We must first feed our text data into a Doc2Vec model to train it. We can obtain these representation vectors by applying this model to our reviews.

Step 4:
The Term Frequency - Inverse Document Frequency (TF-IDF) values for each word and document are then added.

However, why not just count the number of times each word appears in each document? The issue with this strategy is that it doesn't consider the general significance of words in the texts. It is unlikely that an analysis would yield any useful information from a word that appears in nearly every text. Rare words, on the other hand, might mean a lot more things.

This issue is resolved by the TF-IDF metric:

We add TF-IDF columns for every word that appears in at least 10 different texts to filter some of them and reduce the size of the final output. TF calculates the traditional number of times the word appears in the text; IDF calculates the relative importance of this word based on the number of texts in which it can be found.

After completing all the steps, the dataset looks like this:

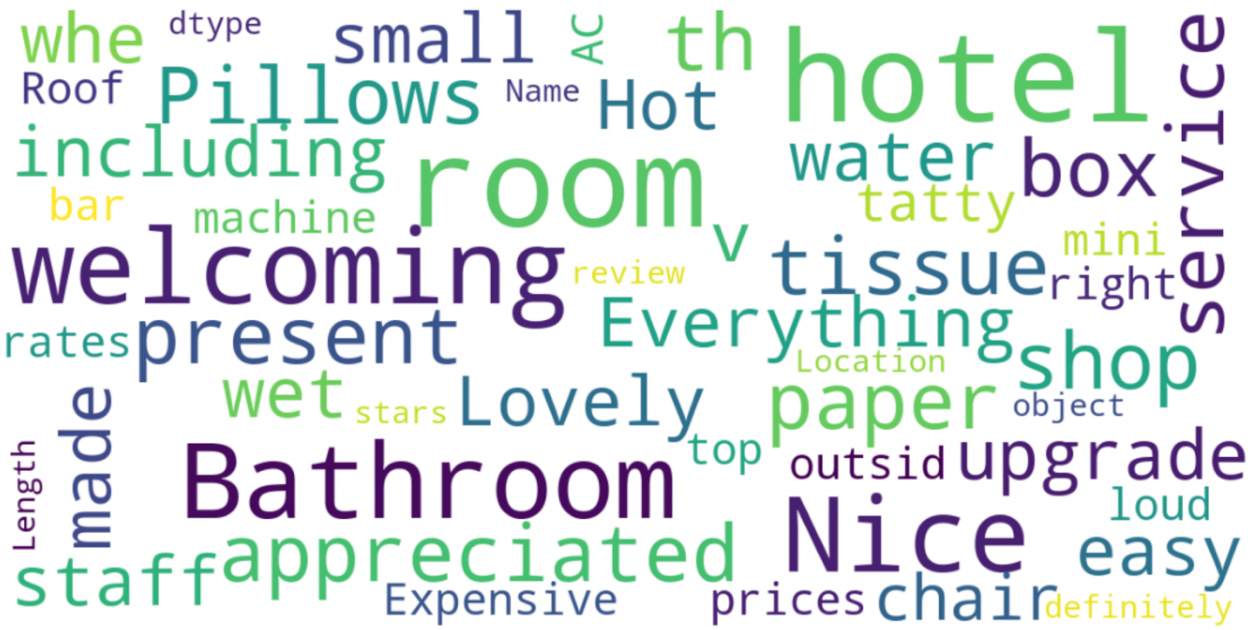
	review	is_bad_review	review_clean	neg	neu	pos	compound	nb_chars	nb_words	doc2vec_vector_0	...	word_yet	word_yoghurt	word_yog
488440	Would have appreciated a shop in the hotel th...	0	would appreciate shop hotel sell drinking wate...	0.049	0.617	0.334	0.9924	599	113	-0.207857	...	0.0	0.0	
274649	No tissue paper box was present at the room	0	tissue paper box present room	0.216	0.784	0.000	-0.2960	44	10	0.099288	...	0.0	0.0	
374688	Pillows Nice welcoming and service	0	pillow nice welcome service	0.000	0.345	0.655	0.6908	36	7	0.072904	...	0.0	0.0	
404352	Everything including the nice upgrade The Hot...	0	everything include nice upgrade hotel revamp s...	0.000	0.621	0.379	0.9153	155	27	0.015668	...	0.0	0.0	
451596	Lovely hotel v welcoming staff	0	lovely hotel welcome staff	0.000	0.230	0.770	0.7717	32	7	0.079473	...	0.0	0.0	

After cleaning and performing sentiment analysis the dataset contains 5 rows and 3840 columns. I wanted to see the review proportion like how much was negative and positive. After doing that the proportion was like that:
Positive 0.956761
Negative 0.043239

We can see the data is not really balanced. The negative review is less than 5%.

Data Visualization:

Most of the words refer to hotels: room, staff, breakfast, and so on. Some terms are more relevant to the customer's hotel stay experience: perfect, adored, pricey, detestable, etc.

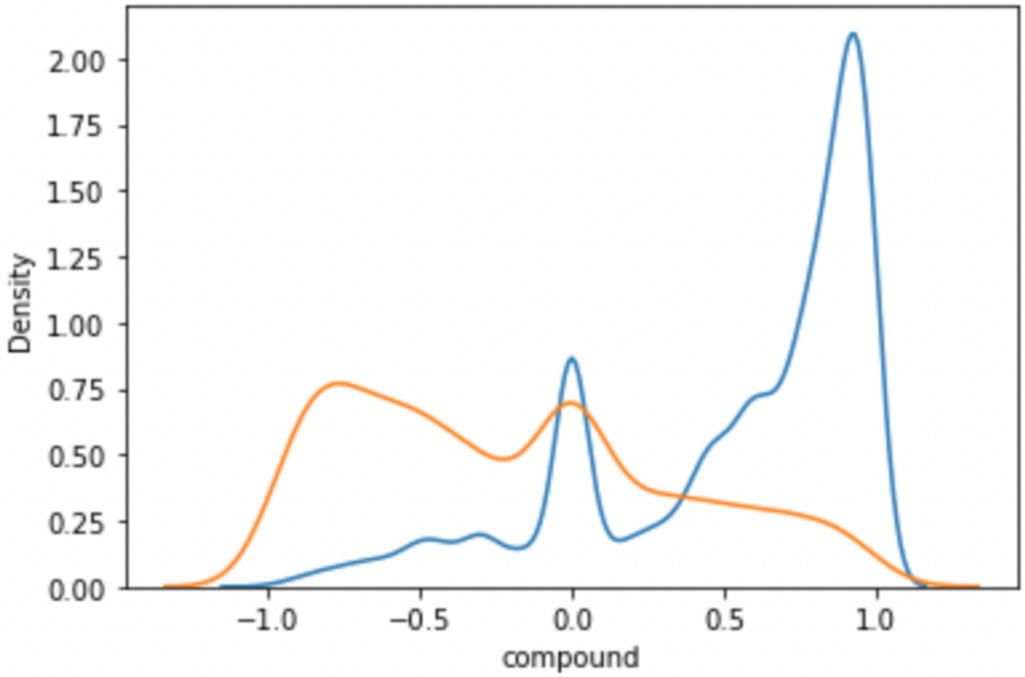


	review	neg
193086	No dislikes LOCATION	0.831
356368	Nothing Great helpful wonderful staff	0.812
318516	A disaster Nothing	0.804
458794	Nothing Excellent friendly helpful staff	0.799
29666	A bit noisy No	0.796
426057	Dirty hotel Smells bad	0.762
263187	Very bad service No	0.758
443796	Nothing perfect	0.750
181508	Window blind was broken	0.744
175316	Nothing Super friendly staff	0.743

Highest negative review

	review	pos
43101	A perfect location comfortable great value	0.931
211742	Clean comfortable lovely staff	0.907
175551	Friendly welcome Comfortable room	0.905
365085	Good location great value	0.904
109564	Clean friendly and comfortable	0.902
145743	Good value amazing location	0.901
407590	breakfast excellent Clean comfort	0.899
407546	Great place I enjoyed	0.881
218571	Beautiful Quirky Comfortable	0.878
436901	Lovely comfortable rooms	0.877

Highest positive review



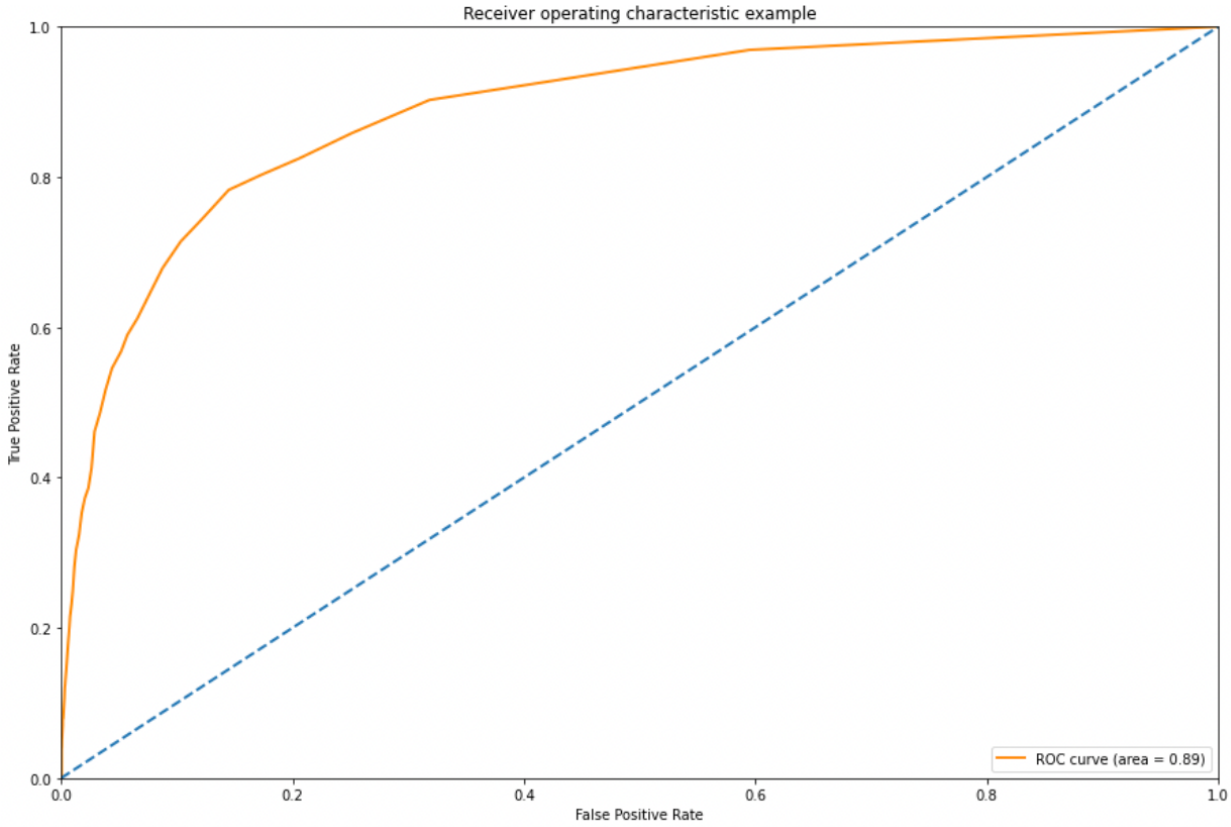
The distribution of positive and negative reviews' sentiments is shown in the graph. We can see that Vader views the majority of favorable reviews as extremely favorable. Going against the norm, terrible audits will generally have lower compound opinion scores.

I choose which features want to use to train our model. Then splitting data into two parts:

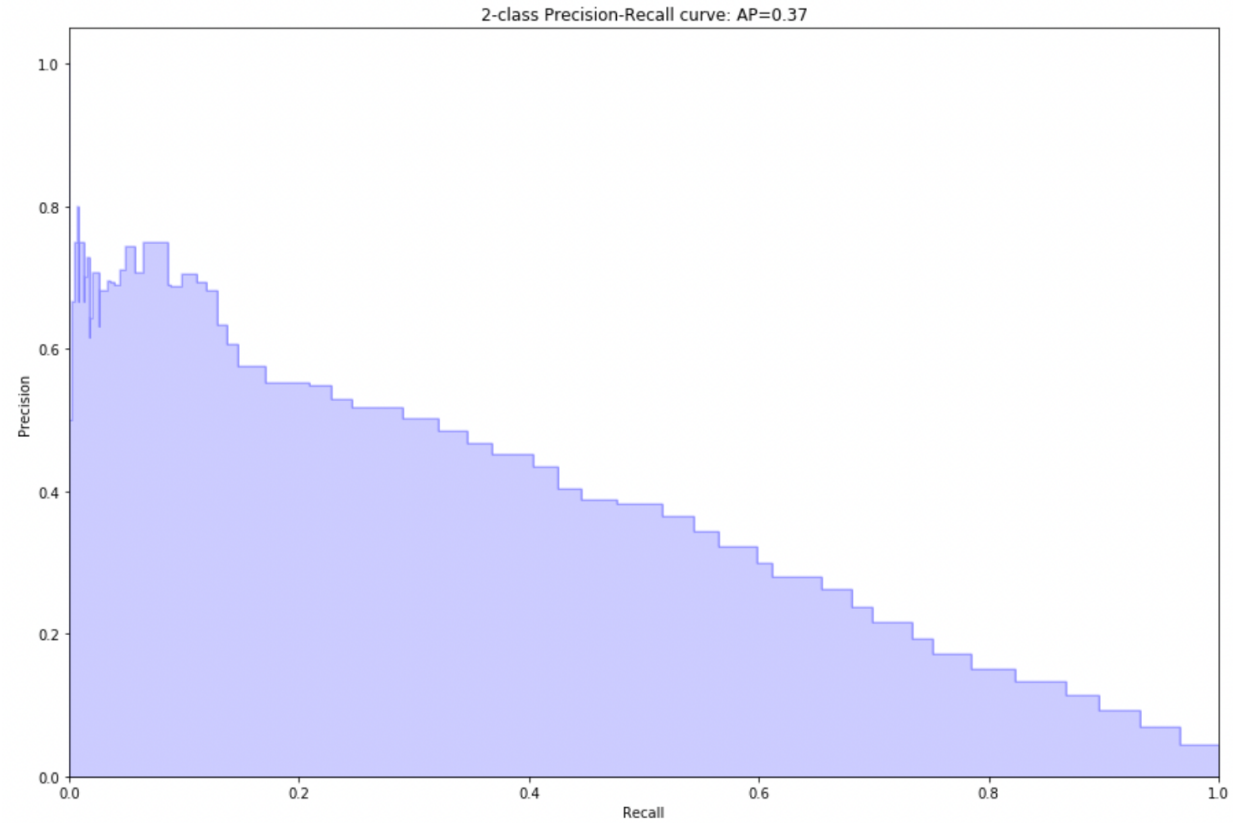
one to train model
one to assess performances

I have used Random Forest (RF) classifier for predictions.
Indeed, the previous sentiment analysis yields the most significant features. Our training also relies heavily on the texts' vector representations. A few words seem to have a genuinely decent significance too.

	feature	importance
3	compound	0.038291
2	pos	0.025844
6	doc2vec_vector_0	0.024596
0	neg	0.021564
10	doc2vec_vector_4	0.018204
8	doc2vec_vector_2	0.017828
7	doc2vec_vector_1	0.017628
9	doc2vec_vector_3	0.017104
4	nb_chars	0.016668
1	neu	0.014248
5	nb_words	0.013682
950	word_dirty	0.009970
2853	word_room	0.009617
2239	word_nothing	0.009399
285	word_bad	0.008822
3216	word_star	0.006675
1945	word_location	0.006579
3202	word_staff	0.006457



The ROC curve is usually a good graph to summarize the quality of our classifier. The higher the curve is above the diagonal baseline, the better the predictions.



As the recall increases, the precision decreases, as can be seen. This demonstrates that we must select a custom-tailored prediction. On the off chance that we want to have a high review, we ought to set a low forecast threshold that will permit us to recognize the vast majority of the perceptions of the positive class, however with a low accuracy. On the other hand, if we want to be very sure of our predictions but don't mind not finding all positive observations, we should set a high threshold, which will give us high precision and low recall.

Conclusion:

In conclusion, the analysis of hotel reviews using sentiment analysis techniques provides valuable insights for individuals who are considering opening a hotel business. By analyzing customer sentiments, hotel owners can better understand what aspects of their service are most important to their customers and how they can improve upon them. Additionally, sentiment analysis can help hotel owners identify potential problem areas before they become significant issues, allowing for proactive steps to be taken to prevent negative customer experiences.

Overall, sentiment analysis provides an excellent tool for hotel owners to better understand their customers and to make data-driven decisions that can lead to a more successful hotel business. However, it is important to keep in mind that sentiment analysis is only one part of the equation and should be used in conjunction with other research and analyses to make the most informed business decisions.

Bibliography:

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168-177.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135.

Mishra, R., & Biswas, P. (2018). Sentiment analysis of hotel reviews using machine learning techniques. Procedia Computer Science, 132, 1263-1272.