Unstructured data analysis

Darya Kandalina

Introduction

In today's world, when the Internet has become the main source of information for people, it is important for people to get the most relevant and useful information from it before making any decisions. So, for example, when planning a vacation, people first of all prefer to study the description of hotels, look at the proposed photos, and, most importantly, study hotel reviews before making their choice. TripAdvisor estimates that 81% of people often or always read reviews before booking a hotel. This statistic proves that customer reviews play an important role in the hospitality industry. In this context, sentiment analysis (SA) of hotel reviews is becoming increasingly important. Each review reflects certain sentiments, positive or negative, that define a hotel's reputation.

That is why, for each company, the issue of collecting and analyzing reviews is very relevant. Companies around the world rely on reviews to better understand customer satisfaction. Most review platforms use a "scaled rating system" such as Amazon's "star ratings" or Youtube's "like or dislike" system, however, many companies do not have such rating systems. So by categorizing different hotel reviews on Tripadvisor, we can determine the sentiment for each review, which can help both consumers and business owners understand how valuable a product or service is.

In this paper, we will propose several models for classifying reviews, compare them with each other, and in addition, for reviews with an average or negative rating, we will identify the main points that did not suit customers - this can be a growth point for hotels.

Literature Review

Sentimental analysis is used to analyze text reviews. Sentiment analysis, also called opinion analysis, is a natural language processing (NLP) approach that detects the emotional tone of a text. It is a popular way for organizations to define and categorize opinions about a product, service, or idea. Sentiment analysis involves the use of data mining, machine learning (ML), artificial intelligence, and computational linguistics to extract sentiment and subjective information from text, such as whether it expresses positive, negative, or neutral feelings.

Sentiment analysis uses machine learning models to analyze human language text. The metrics used are designed to determine whether the overall mood of a piece of text is positive, negative, or neutral.

Types of Sentiment Analysis

Sentiment analysis systems fall into several categories:

The detailed sentiment analysis breaks down sentiment indicators into more precise categories such as very positive and very negative. This approach is similar to rating opinions on a scale of one to five stars. Thus, this approach is effective in assessing customer satisfaction.

Emotion detection analysis identifies emotions, not positives and negatives. Examples include happiness, disappointment, shock, anger, and sadness.

Intent-based analysis recognizes text motifs in addition to opinion. For example, an online comment expressing frustration about a battery replacement might have the intention of contacting customer support to resolve the issue.

Aspect analysis examines a particular component that is mentioned positively or negatively. For example, a customer might write a review about a product saying that the battery life is too short. The sentiment analysis system will notice that the negative attitude is not related to the product as a whole, but to battery life.

Sentimental analysis is a rather complex process and can lead to the following problems:

- Neutral feelings. Neutral sentiment comments tend to create problems for systems and are often misidentified.
- Incomprehensible language. Mood is difficult to determine when systems don't understand context or tone. Answers to questions such as "nothing" or "everything" are difficult to classify unless the context is given; they can be marked as positive or negative depending on the question. This is known as lexical ambiguity.

The algorithms have trouble resolving a pronoun that refers to something that comes before a pronoun in a sentence. For example, when analyzing the comment “We went for a walk and then had dinner. I didn’t like it”, the system may not be able to determine whether the writer didn’t like a walk or dinner.

- Unclassified language. Computer programs have a hard time understanding emoticons and unnecessary information. Special attention should be paid to training models with emoji and neutral data so that they do not label texts incorrectly.
- Ambiguous feelings. People can be contradictory in their statements. Most reviews will have both positive and negative comments. This situation can be dealt with by analyzing sentences one at a time. However, sentences containing two contradictory words, also known as contrastive conjunctions, can confuse sentiment analysis tools. For example, "The packaging was terrible, but the product was great."
- Small datasets. Sentiment analysis tools work best when analyzing large amounts of textual data. Smaller datasets often do not provide the necessary information.
- Language evolution. Language is constantly changing, especially on the Internet, where users are constantly creating new abbreviations, acronyms, and using bad grammar and spelling. This level of variation and evolution can be difficult for algorithms.
- Fake reviews. Algorithms cannot always distinguish between real product reviews and fake or other pieces of text generated by bots.

There are various algorithms that can be implemented in sentiment analysis models, depending on how much data needs to be analyzed and how accurate the model needs to be.
Sentiment analysis algorithms fall into one of three categories:

- Rule Based: These systems automatically perform sentiment analysis based on a set of manually created rules.
- Automatic: Systems rely on machine learning techniques to learn from data.
- Hybrid systems combine both rule-based and automated approaches.

Rule-based Approaches

Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity, polarity, or the subject of an opinion.

These rules may include various NPL methods developed in computational linguistics, such as:

- Stemming, tokenization, part-of-speech tagging and parsing.
- Lexicons (i.e. lists of words and expressions).

Here's a basic example of how a rule-based system works:

1. Defines two lists of polarized words (e.g. negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc).
2. Counts the number of positive and negative words that appear in a given text.
3. If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment.

Rule-based systems are very naive since they don't take into account how words are combined in a sequence. Of course, more advanced processing techniques can be used, and new rules added to support new expressions and vocabulary. However, adding new rules may affect previous results, and the whole system can get very complex. Since rule-based systems often require fine-tuning and maintenance, they'll also need regular investments.

Automatic Approaches

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on ML techniques. A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, e.g. positive, negative, or neutral.

Text data classification

As mentioned earlier, in order to classify text data, it is first necessary to process the text and perform tokenization. After that, various machine learning models can be applied, both classical and special neural networks created for NLP.

Next, we will dwell on some classification models in more detail.

Naive Bayes classifier

Naive Bayes is a simple algorithm that classifies text based on the probabilities of occurrence of events. The algorithm is based on the Bayes theorem, which helps to find the conditional probabilities of events that have occurred based on the probabilities of occurrence of each individual event. Using the Bayes equation, the probability is calculated for each class with the corresponding sentences. Based on the probability value, the algorithm decides whether the sentence belongs to one or the other.

KNN

KNN - K nearest neighbor. It is a machine learning algorithm that classifies new text by matching it to the closest matches in the training data to make predictions. Since the neighbors have similar behavior and characteristics, they can be attributed to one. Similarly, the KNN algorithm determines the K nearest neighbors by the degree of closeness between the training data. The model is trained in such a way that when new data is passed through the model, the text is easily matched to the group or class to which it belongs.

Decision trees

Decision trees reproduce logical schemes that allow you to get the final decision on the classification of an object by answering a hierarchically organized system of questions. Moreover, the question asked at the next hierarchical level depends on the answer received at the previous level.

To work with text classification, the simplest and most popular link is TF-IDF + one of the classic models, in particular those described above. This process requires a small amount of computational resources, but has its drawbacks, in particular, it does not take into account the context of the sentence. In addition, the process of using such a bundle requires additional operations: cleaning, lemmatization. There are transformer algorithms that allow you to skip this step, for example, BERT, which is also able to understand the context of words, which is its main advantage.

BERT is a neural network based on the composition of transformer encoders. BERT is an autoencoder. Each layer of the encoder uses two-way attention, which

allows the model to take into account the context on both sides of the token in question, and therefore more accurately determine the values of the tokens.

When text is supplied to the network input, it is first tokenized. Tokens are words available in the dictionary, or their constituent parts - if the word is not in the dictionary, it is split into parts that are present in the dictionary. The dictionary is a component of the model - for example, BERT-Base uses a dictionary of about 30,000 words. In the neural network itself, tokens are encoded by their vector representations (embeddings), namely, the representations of the token itself (pre-trained), its offer number, and the position of the token inside its offer are connected. The input data is received and processed by the network in parallel, not sequentially, but the information about the mutual arrangement of words in the original sentence is stored, being included in the positional part of the embedding of the corresponding token.

The output layer of the main network has the following form: a field responsible for the answer in the task of predicting the next sentence, as well as tokens in an amount equal to the input. The reverse transformation of tokens into the probability distribution of words is carried out by a fully connected layer with the number of neurons equal to the number of tokens in the original dictionary.

BERT is trained simultaneously on two tasks - predicting the next sentence and generating a missing token (masked language modeling). The BERT input is tokenized pairs of sentences in which some tokens are hidden. Thus, thanks to the masking of tokens, the network learns a deep bidirectional representation of the language, learns to understand the context of the sentence.

Tasks and expected results of the project

As a result of this study, the following points are expected to be achieved:

- collection and preprocessing of data;

- exploratory data analysis;

- building classification models and comparing them with each other in order to determine the model that can best classify feedback into one of three categories: negative, neutral and positive.

Description and data collection

As data for the study, we will use reviews on one of the largest aggregators - TripAdvisor. We will consider hotels in Thailand, Phuket region.

The data will be self-collected, The start page is the page located at the link: https://www.tripadvisor.com/Hotels-g293920-oa30-zfc4,5-Phuket-Hotels.html

Due to the fact that we collect data on our own and have some time limits for parsing and access limits for the API of the site itself (loading is blocked if the volume is large), we will collect only hotel reviews from the first 2 search pages.

Some hotels have a huge number of reviews, in order to somehow balance them, a limit of 155 reviews was set for each of the hotels.

Data to be collected:

- the name of the hotel

- review text

- date of writing the review

- link to review

- score from review

- the average rating of the hotel on the site.

Example of the collected data:

```
In [30]: df
```

Out[30]:

	hotel_name	review_text	review_data	link	page	review_rank_new	hotel_rank_new
0	Pamookkoo Resort	I HAD A GREAT STAYING WITH THEM DURING MY VISI...	xan l wrote a review Feb 15	https://www.tripadvisor.com//Hotel_Review-g121...	0.0	5.0	4.5
1	Pamookkoo Resort	Great place to stay! Everything is within walk...	poean wrote a review Dec 2022	https://www.tripadvisor.com//Hotel_Review-g121...	0.0	5.0	4.5
2	Pamookkoo Resort	This is by far the worst hotel i have stayed i...	Eva R wrote a review Feb 11	https://www.tripadvisor.com//Hotel_Review-g121...	0.0	1.0	4.5
3	Pamookkoo Resort	We are a family of 4, 2 adults and 2 teenagers...	ClareJens wrote a review Jan 2023	https://www.tripadvisor.com//Hotel_Review-g121...	0.0	2.0	4.5
4	Pamookkoo Resort	Heaven for family traveller's with kids.. it w...	RohitC1409 wrote a review Jan 2023	https://www.tripadvisor.com//Hotel_Review-g121...	0.0	5.0	4.5
...	...	...	...	...	...	...	...
8993	The Shore at Katathani	It is in Phuket where we have discovered this ...	Kuk wrote a review Oct 2021	https://www.tripadvisor.com//Hotel_Review-g784...	30.0	5.0	4.5
8994	The Shore at Katathani	Let's start with Phuket, Thailand where I have...	Ms Gem wrote a review Oct 2021	https://www.tripadvisor.com//Hotel_Review-g784...	30.0	5.0	4.5
8995	The Shore at Katathani	We have stayed 14 Days and enjoyed each one of...	marecker wrote a review Oct 2021	https://www.tripadvisor.com//Hotel_Review-g784...	30.0	5.0	4.5

Thus, we have collected a dataset of almost 9 thousand rows.

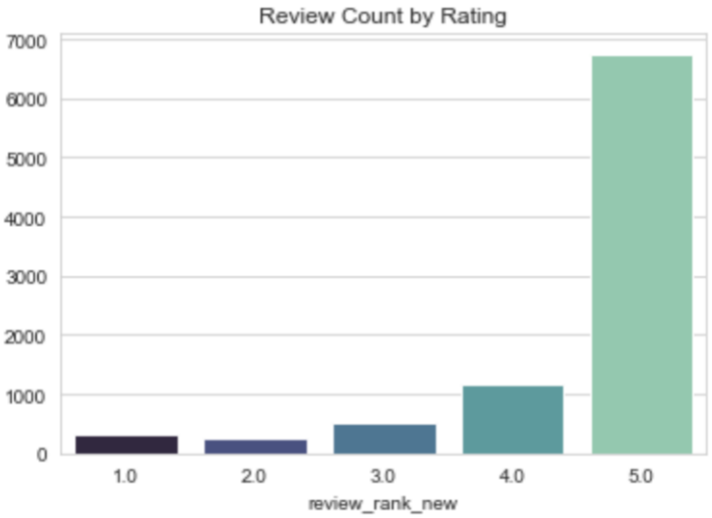
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8998 entries, 0 to 8997
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       8998 non-null   int64  
 1   hotel_name       8998 non-null   object 
 2   review_text      8998 non-null   object 
 3   review_data      8998 non-null   object 
 4   link             8998 non-null   object 
 5   page             8998 non-null   float64
 6   review_rank_new  8998 non-null   float64
 7   hotel_rank_new   8998 non-null   float64
 8   new_rating       8998 non-null   object 
```

There are no missing values in the dataframe.

Exploratory data analysis

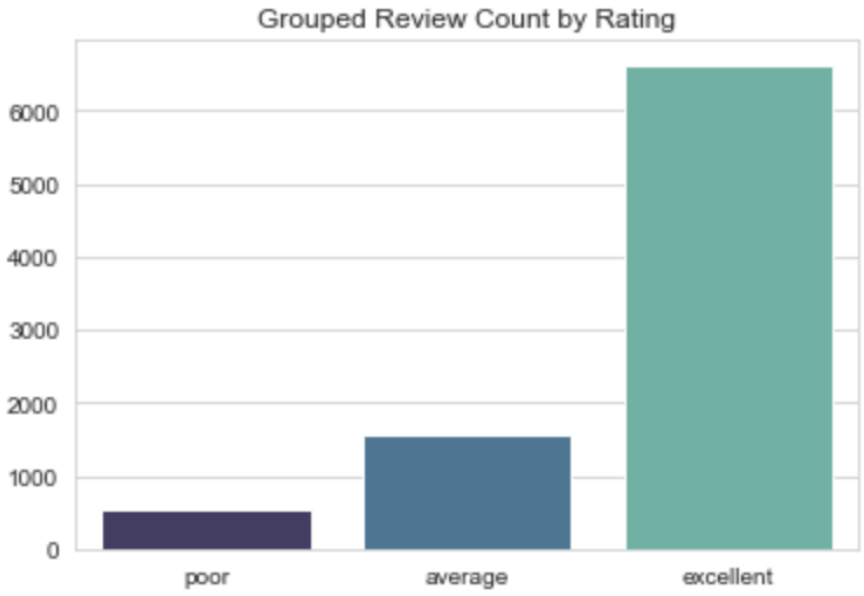
Distribution of reviews by average rating:



We have a strong imbalance, but this is to be expected, because we collected data on 4 and 5 star hotels, as a rule, reviews on them are biased towards good ones.

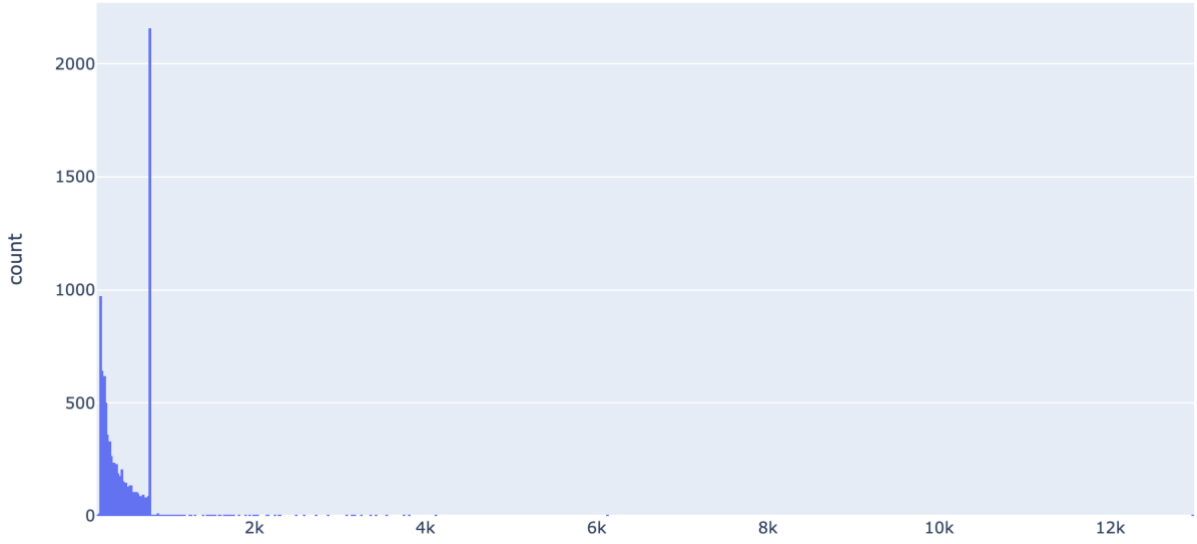
```
The mean rating is 4.496110246721494
```

In view of the fact that in our task we simplify the classification model and divide the reviews into 3 classes instead of 5 (ratings 1 and 2: poor, 3 - average, 5 - excellent), let's see how the reviews are distributed in new classes:

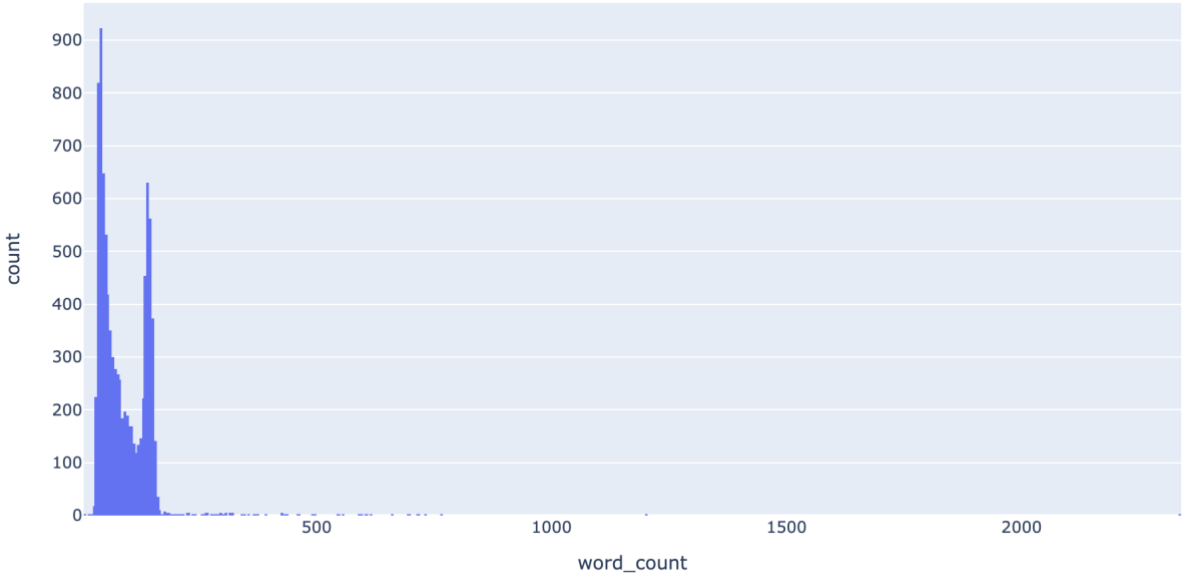


new_rating
average 1673
excellent 6754
poor 571

distribution by length of reviews:

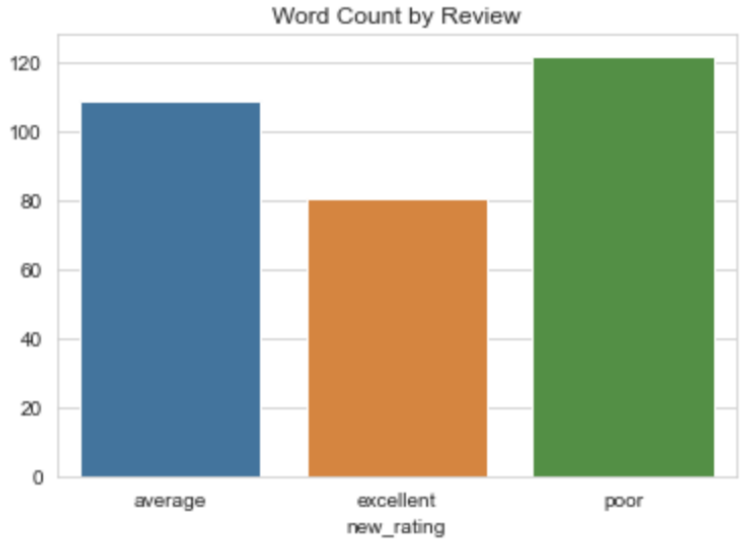


We see a jump within 780-799 characters, this is due to the length limit on the site. Let's see how the reviews are distributed by the number of words:



There were quite a few people who would like to leave long reviews.

Now let's see if the average number of words changes depending on the review class:

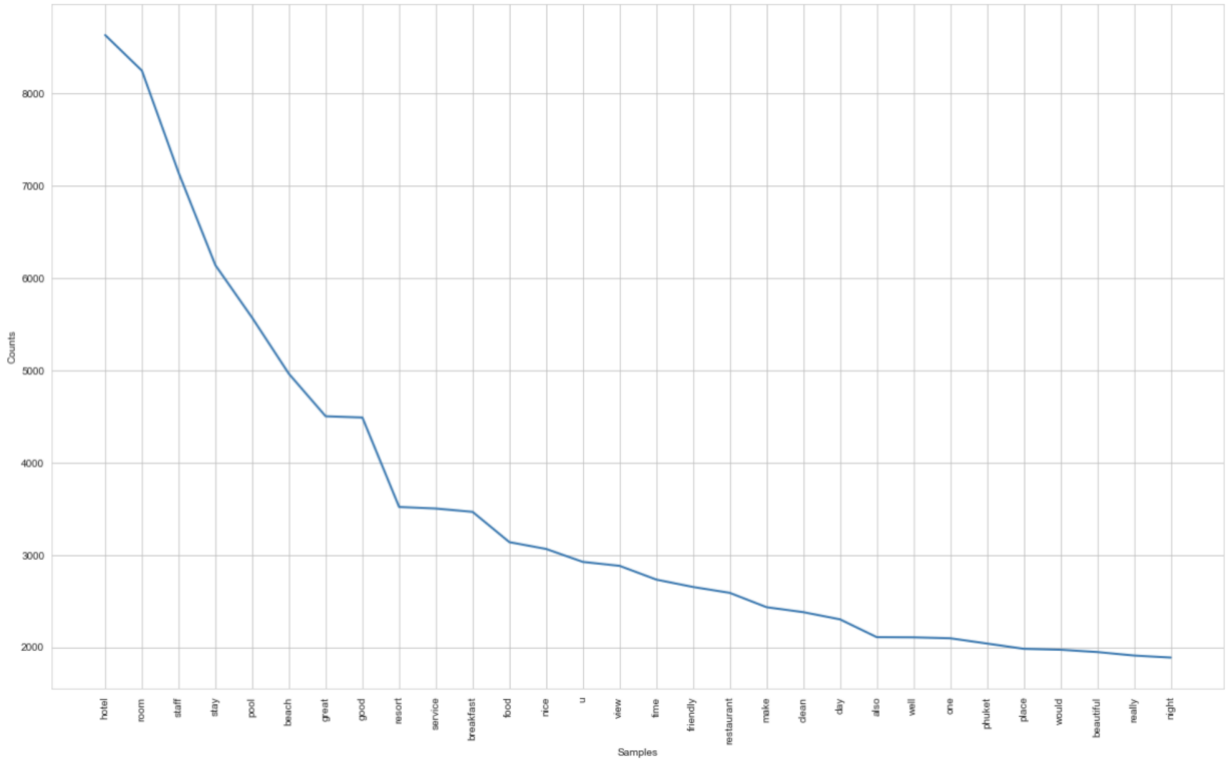


We see that the most detailed reviews are left by those people who give low ratings. This suggests that they try to describe in detail the shortcomings, while people who rate "excellent" leave less detailed (shorter) reviews.

Primary preprocessing

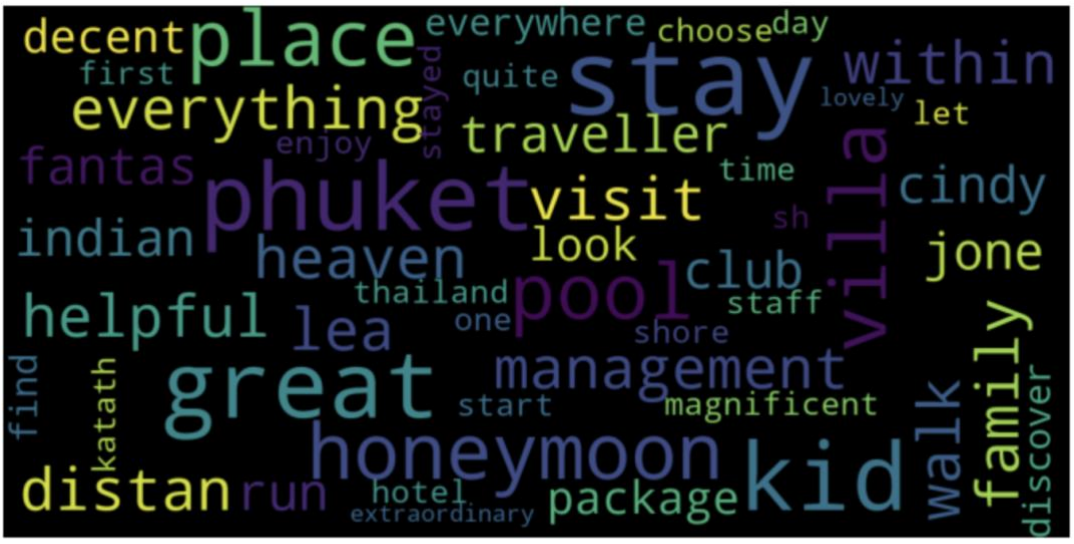
For a more detailed analysis of reviews, it is necessary to bring reviews to a single style, for which we created a function that performs lemmatization, brings everything to the same case and removes punctuation and symbols. Tokenization has been performed.

After applying this function, let's see which words are most often found in reviews:



It is expected that the word hotel is found in almost every review. Also often people mention the facilities, the staff, the pool, the beach, the service. Indeed, these points are the main criteria by which to describe the hotel.

Let's compare word clouds depending on the review class:



excellent

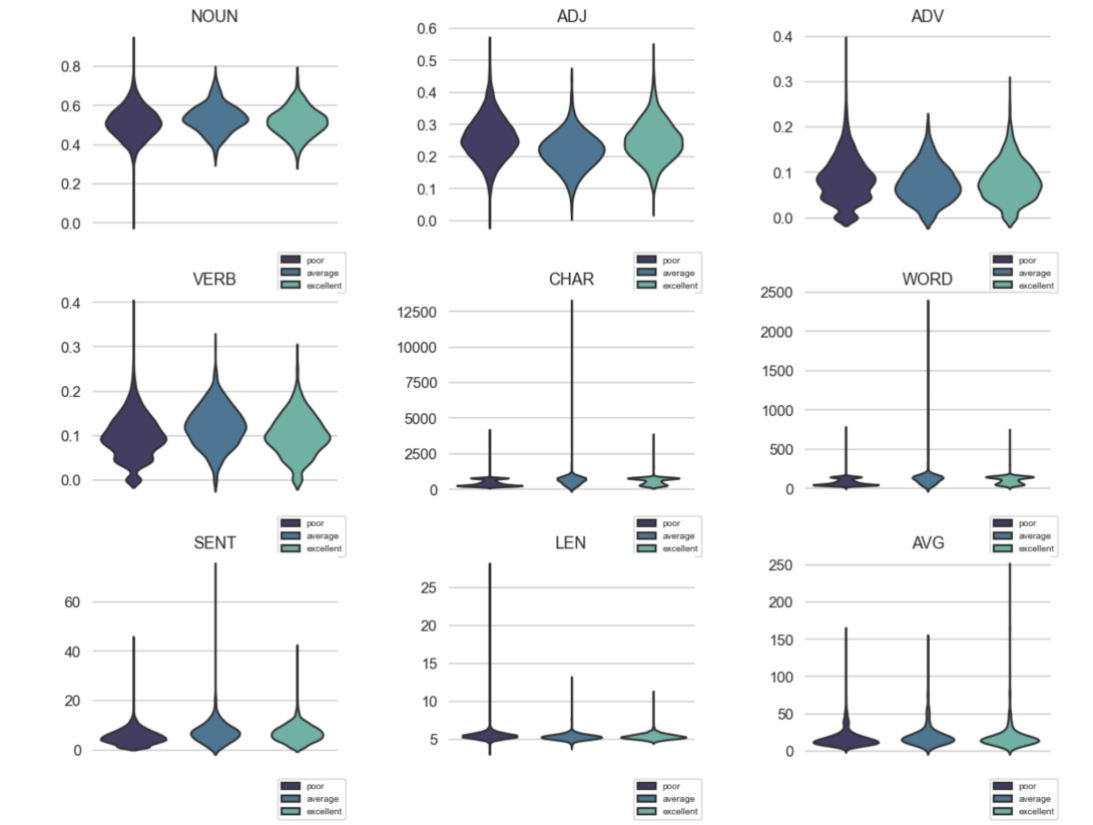
average

poor

We see how the set of adjectives found in reviews changes depending on what rating it has. So, for great reviews, we see: helpful, great, magnificent. While for the poor: leaky, bad, dirty.

Comparison of the recall structure class by class.

Let's see if the parts of verbs, nouns and adjectives differ for different classes of comments.

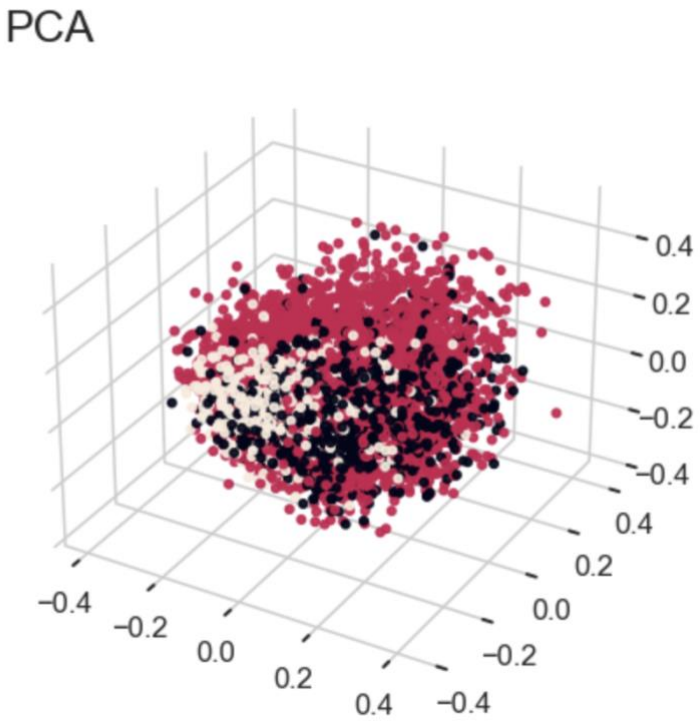


Where
NOUN/ ADJ/ ADV/ VERB – percentage of nouns/ adjectives/ adverbs/ verbs
CHAR - number of characters
WORD - number of words
SENT - number of sentences
LEN - average word length
AVG - average sentence length

In terms of percentages, there is no strong obvious difference, the poor almost everywhere have more elongated "tails", that is, more emissions, but this is probably due to the small proportion of this class compared to others.

Model training results
Normalization of a tokenized vector

The TF-IDF Vectorizer is utilized, which takes into account how frequent a word appears in a document and also how unqiue the word is in overall corpus. In order to capture the most meaningful words, we cut off the top 10% and bottom 5% of words in the documents.
We used PCA to reduce the multidimensionality of some of the models to determine if the results would be different.
The training and training samples are divided in the proportion of 0.8 and 0.2

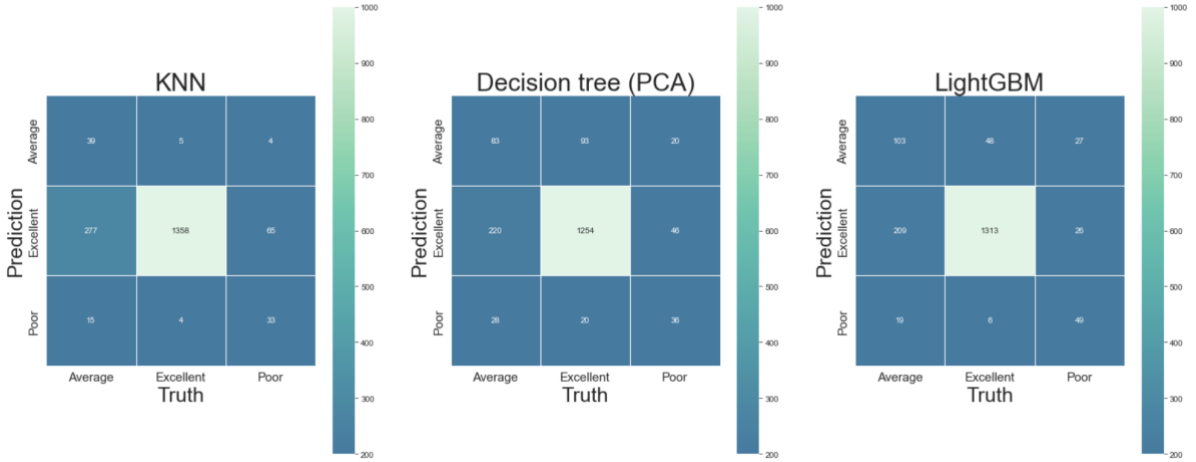


Accuracy and F1-score were chosen as the main metrics by which models will be compared [10].

In order to achieve the best quality from each of the models, it is necessary to select hyperparameters, for which the grid search algorithm was used in the work.

name	accuracy score	f1 score
Naive Bayes	0.767778	0.678534
Logistic Regression	0.697222	0.724605
Logistic Regression (PCA)	0.702222	0.728622
Decision Tree	0.746111	0.711705
Decision Tree (PCA)	0.762778	0.739605
Random Forest	0.760556	0.658438
Light GBM	0.813889	0.790127
KNN	0.794444	0.734661

Boosting trained by Light GBM proved to be the best. In terms of accuracy, the unsupervised model, KNN, also became a strong model, however, in terms of F1-score, it is weaker than decision trees using PCA, which is third in accuracy. Confusion matrix for these three algorithms:



It turns out that Light GBM most well separates weak and medium reviews. However, it is interesting that the unsupervised method also performed quite well. This shows that companies have the ability not to ask for a rating and write a review at the same time, for fear of their inconsistency with each other, but to use clustering to rate based on the review.

Conclusion

Summing up, I would like to note that in the course of the project, a study was carried out on the methods of semantic analysis of texts, data were collected independently, their exploratory analysis was performed, words were found that distinguish bad reviews from good ones, after which several models were trained in conjunction with TF-IDF.

As a result of the work, we were able to choose a model - Light GBM, which has the smallest number of false positives and is the best in terms of quality metrics. This work can be useful for automatic grading according to the text of the review.

To further improve the performance, you should try the following:

- train the transformer model and compare the results
- explore the relationship with the age of the review on the site and the average rating for the hotel
- apply algorithms that could illustrate the specific points that users complain about.

List of references

1.“Bert (Языковая Модель).” BERT (Языковая Модель) - Викиконспекты, https://neerc.ifmo.ru/wiki/index.php?title=BERT_%28%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%29.

2.Brownlee, Jason. “4 Types of Classification Tasks in Machine Learning.” MachineLearningMastery.com, 19 Aug. 2020, https://machinelearningmastery.com/types-of-classification-in-machine-learning/.

3.Editor. “Sentiment Analysis in Hotel Reviews: Developing a Decision-Making Assistant for Travelers.” AltexSoft, AltexSoft, 12 Mar. 2021, https://www.altexsoft.com/blog/sentiment-analysis-hotel-reviews/.

4.“Getting Started with Sentiment Analysis Using Python.” Hugging Face – The AI Community Building the Future., https://huggingface.co/blog/sentiment-analysis-python.

5.Gupta, Saurabh. “Popular Classification Models for Machine Learning.” Analytics Vidhya, 2 Dec. 2020, https://www.analyticsvidhya.com/blog/2020/11/popular-classification-models-for-machine-learning/.

6.Li, Susan. “Latent Semantic Analysis & Sentiment Classification with Python.” Medium, Towards Data Science, 6 Dec. 2018, https://towardsdatascience.com/latent-semantic-analysis-sentiment-classification-with-python-5f657346f6a3.

7.Niggl, Dennis. “Sentiment Analysis on TripAdvisor Hotel Reviews with Python and NLP.” Medium, Python in Plain English, 15 July 2022, https://python.plainenglish.io/sentiment-analysis-on-tripadvisor-hotel-reviews-with-python-and-nlp-68b3555d816c.

8.Team, Towards AI Editorial. “Understanding Semantic Analysis Using Python - NLP.” Medium, Towards AI, 13 May 2021, https://pub.towardsai.net/understanding-semantic-analysis-using-python-nlp-f48016422677.

9.Victor, Tan Boon Kiat. “Natural Language Processing on Hotel Reviews.” Medium, Medium, 6 Jan. 2020, https://medium.com/@victortantp/natural-language-processing-on-hotel-reviews-b71580aaa6c3.

10.“Which Metrics Are Used to Evaluate a Multiclass Classification Model's Performance?” For Predictions in Minutes. No Code Required., https://www.pi.exchange/knowledgehub/metrics-to-consider-when-evaluating-a-multiclass-classification-models-performance.