Sentiment Analysis on 515k Hotel Reviews

Xin Qu ([xinq2@illinois.edu](mailto:xinq2@illinois.edu)), Biruo Zhao ([biruoz2@illinois.edu](mailto:biruoz2@illinois.edu))

# Background

Tourism industry has transitioned from a brick-and-mortar and person-to-person business into a digital savvy and omnipresent travel service network. Taking booking.com for example, as one of the most successful Online Travel Agency (OTA), its website and mobile apps are available in over 40 languages, offer 28,988,780 total reported listings, and cover 144,354 destinations in 229 countries and territories worldwide. Every day, more than 1,550,000 room nights are reserved on this platform. Therefore, how to retrieve, analyze, and categorize those emotional and experiential elements of tourist activities and capitalize on those digital footprints has become a great challenge and major concern of tourism businesses. Sentiment Analysis comes into the rescue, Sentiment Analysis basically refers to the use of computational linguistics and natural language processing to analyze text and identify its subjective information (Alaei, Becken, & Stantic, 2017). Opinion Mining or Sentiment Analysis is based on the idea of unlocking the hidden value of opinions to achieve deeper understanding of customers’ need and more informed and actional business insights.

# About the Data

This dataset was scraped from Booking.com by Jason Liu and publicly available on Kaggle (<https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe>).

The aim of applying Sentiment Analysis to this dataset is two-fold:

#### **1. for hotel managers:**

Hotels can leverage opinion polarity and sentiment topic recognition to gain a deeper understanding of customers’ feedback and their drivers. Also, to extract customers’ emotional tone from the reviews they posted will help provide good opportunities for hotels to re-evaluate and improve their customer service and products accordingly.

#### **2. for potential customers:**

Sentiment Analysis helps answer questions like if two hotels have the same review score, how can a potential customer find the right hotel?

All customers should no longer be placed in the one bucket but demand different service and products to satisfy certain needs. Some customers have high standards on hotel staff that take care of them and create guest experience. Some guests want spaciousness and quality of facilities to achieve a restful sleep. For some guests, price or good value may be the most key factor in their decision to book. And others are willing to pay extra to indulge themselves with high-end experience in their travel. In attempt to drive right customers to right hotels, we need to find a way to evaluate and sort hotels by different features.

To utilize Sentiment Analysis to its full potential, we bring a trending concept “Aspect-based Sentiment Analysis” into this project. Different from a traditional sentiment classification task which tends to treat an entire entity as a whole, Aspect-based Sentiment Analysis focuses on how to estimate sentiment score for different aspects of an entity, how positive or negative the opinions are on average for each aspect.

# Initial Observation (data\_process.ipynb)

This dataset contains 515,738 customer reviews and score of 1492 luxury hotels across Europe gathered by Bookings.com from August 2015 to August 2017.

All 17 fields in the dataset are described below:

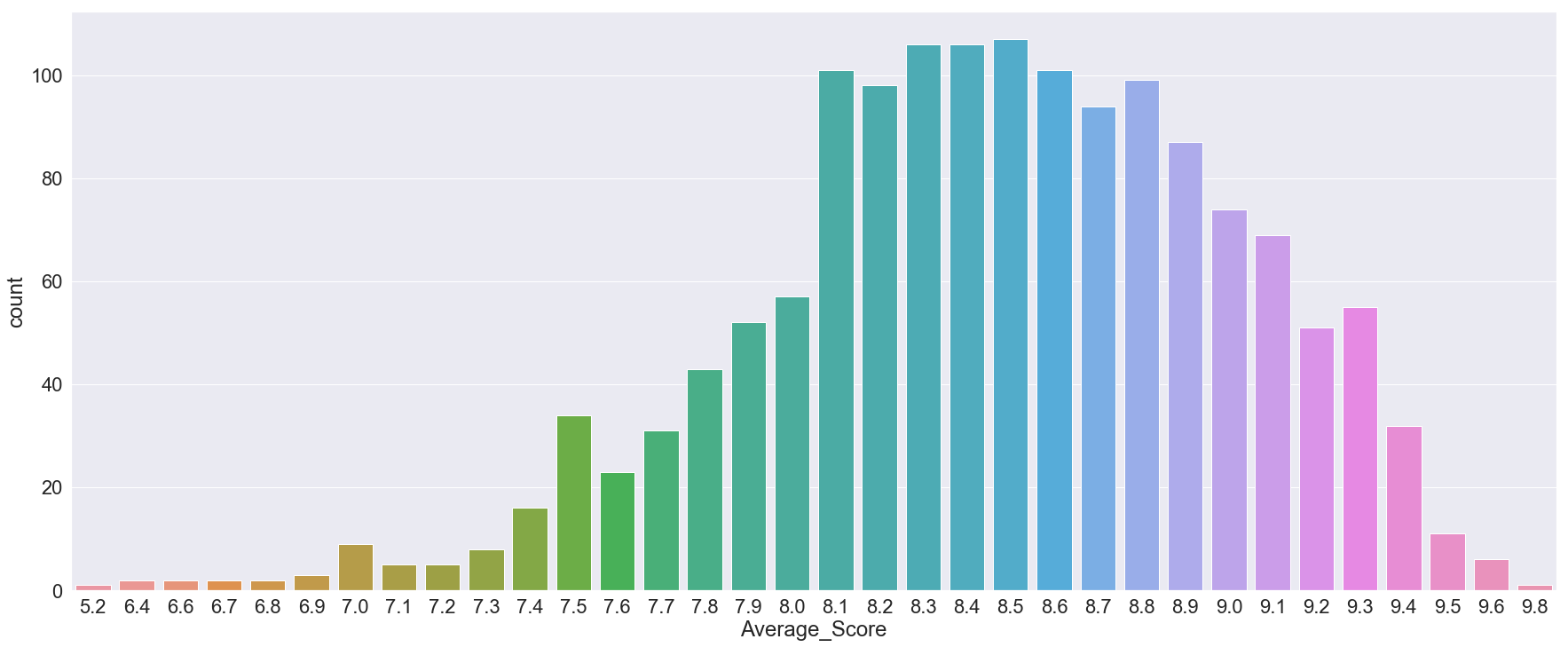
* Hotel\_Address: Address of the hotel.
* Review\_Date: Date when reviewer posted the corresponding review.
* Average\_Score: Average Score of the hotel, calculated based on the latest comments in the last year.
* Hotel\_Name: Name of the hotel.
* Reviewer\_Nationality: Nationality of the reviewer.
* Negative\_Review: Negative/bad things the reviewer wrote about the hotel. If the reviewer didn’t give a negative review, its value shows 'No Negative'.
* Review\_Total\_Negative\_Word\_Counts: Total words in the negative review.
* Positive\_Review: Positive/good things the reviewer wrote the hotel. If the reviewer didn’t give a negative review, its value shows 'No Positive'.
* Review\_Total\_Positive\_Word\_Counts: Total words in the positive review.
* Reviewer\_Score: A score the reviewer has given to the hotel, based on his/her experience.
* Total\_Number\_of\_Reviews\_Reviewer\_Has\_Given: Number of Reviews the reviewer has given in the past.
* Total\_Number\_of\_Reviews: Total number of valid reviews the hotel has.
* Tags: Tags reviewer gave the hotel.
* days\_since\_review: Duration between the review date and scrape date.
* Additional\_Number\_of\_Scoring: There are also some guests who just made a scoring on the service rather than a review. This number indicates how many valid scores without review in there.
* lat: Latitude of the hotel.
* lng: longtitude of the hotel.

We take closer look in Average\_Score, Hotel\_Name, Negative\_Review, Review\_Total\_Negative\_Word\_Counts, Positive\_Review, Review\_Total\_Positive\_Word\_Counts, Reviewer\_Score for Sentiment Analysis. Since both negative reviews and positive reviews are already split and provided by the dataset, so starting with binary classification would be natural choice for this project.

And geographical information including Hotel\_Address, lat, lng might be helpful for data visualization in the later phase of the project.

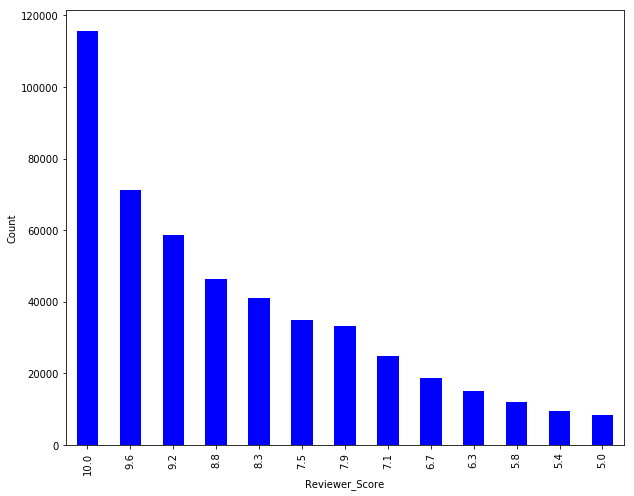
#### Average Score Distribution:

The histogram below represents the distribution of Hotel Average\_Score. It ranges from 5.2 to 9.8, and the most common average score are between 8.1 to 9.1. Since Average\_Score is calculated based on all Reviewer\_Scores from the previous year, it tends to “balance” some extreme values, so we see a relatively smooth and “normal” distribution.

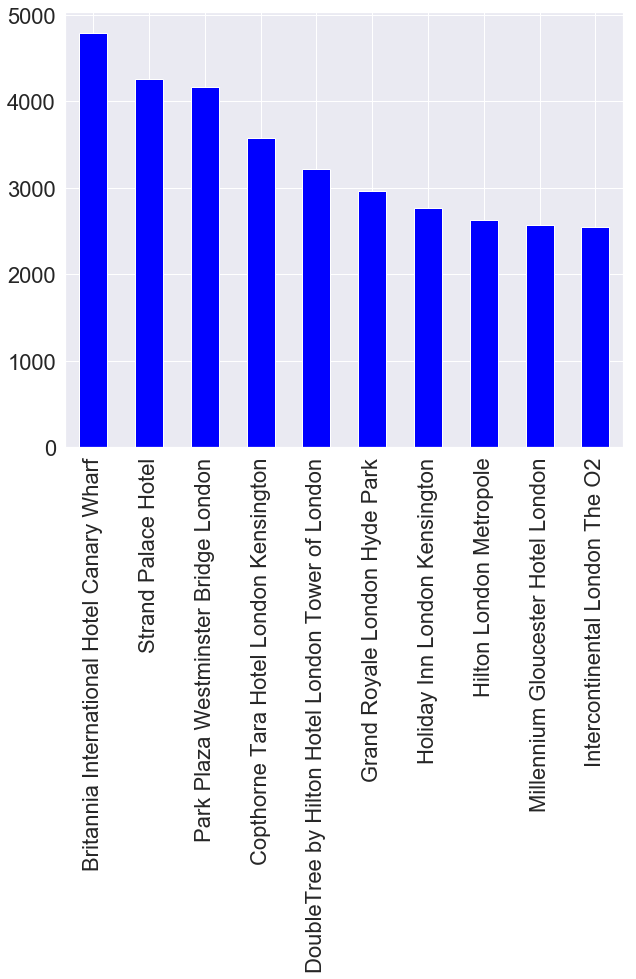


Reviewer Score Distribution:

The histogram below represents the top 5 distribution of Hotel Reviewer\_Score. The range of review\_score is from 1.0 to 10.0.



Top 10 Most Reviewed Hotels:



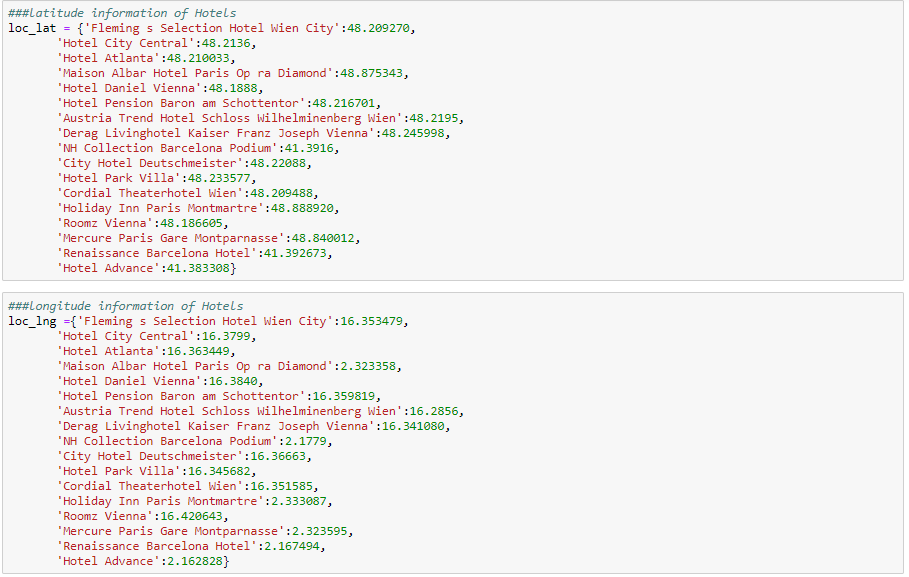
Word could distribution for Review\_Nationality:



# Simple Binary Classification (data\_process.ipynb)

1. Handle duplicated and missing values

We start with removing duplicated values and checking missing values. Most missing values are Hotel\_Names, latitudes and longitudes of hotels, we use information provided by <http://latlong.org/> to manually fill in those blanks for future geographical display.



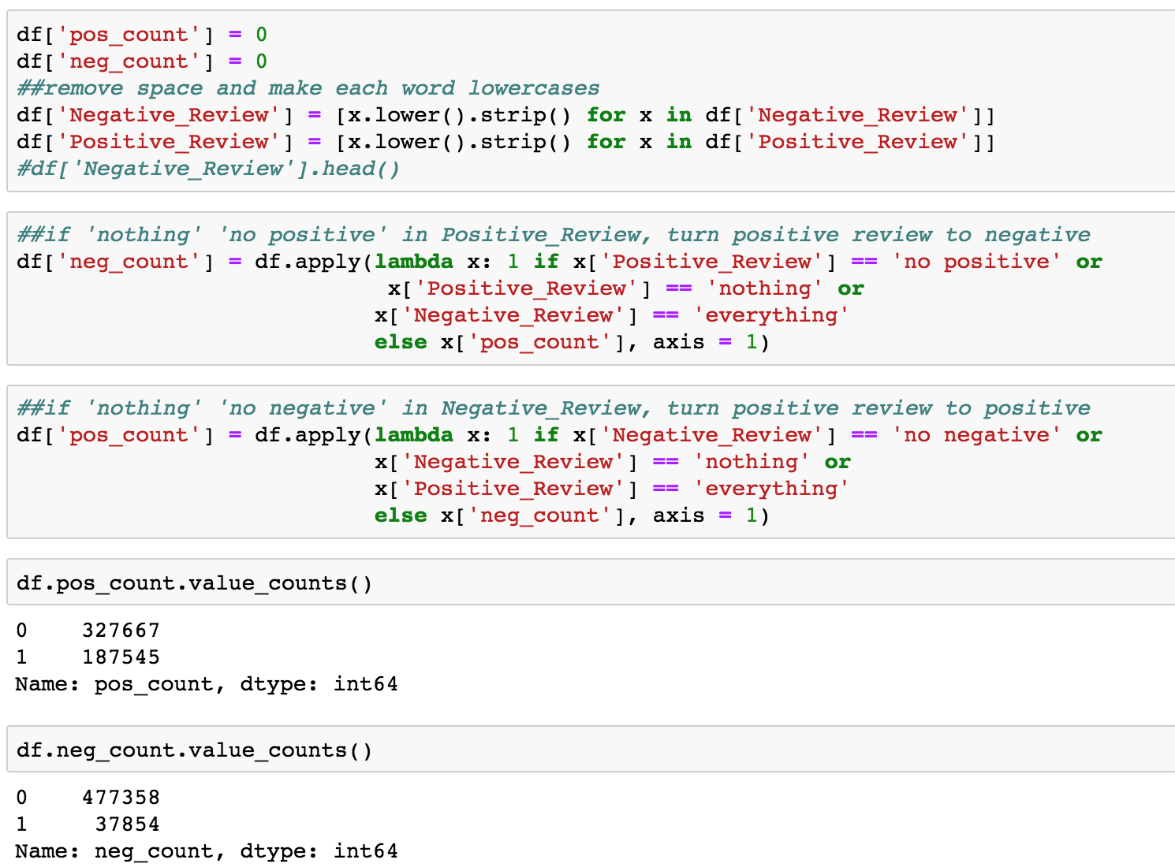
1. Stemming and Tokenization

We install and use **NLTK** libraries to do basic data pre-processing. When apply built-in stopwords in NLTK library, the word “us” shows in the most 20 common words in negative words. In order to remove a stop word such as “us”, apply a pre-downloaded “stopwords.txt” file. Then stem the word by applying the out-of-box PorterStemmer() from NLTK.



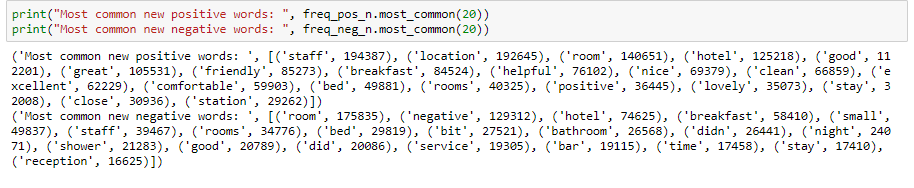
1. Label positive and negative reviews

We clean up the corpus by eliminating white space, removing numbers and coverting all words in reviews to lower case. For simplicity's sake, each positive/negative review is treated as a document to be classified. We assign each positive review numeric value 1 and each negative review numeric value 0. One thing worth noting is we treat “no positive” in positive review as a 100% negative review, “no negative” in negative review as a 100% positive review and treat “nothing” in positive or negative review as negative / positive review respectively. The other case is to treat “everything” in positive/negative review as negative/positive review respectively. Based on these labels, we have prepared 477,358 negative reviews and 327,667 positive reviews for the next step, classifier comparison and evaluation.

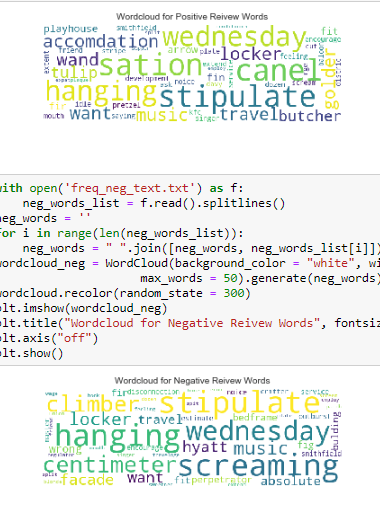


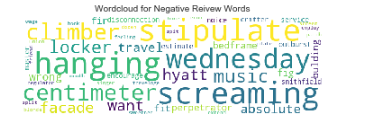


We’re able to extract 20 most common positive words and 20 most common negative words and their frequencies:



Their word clouds are generated below:



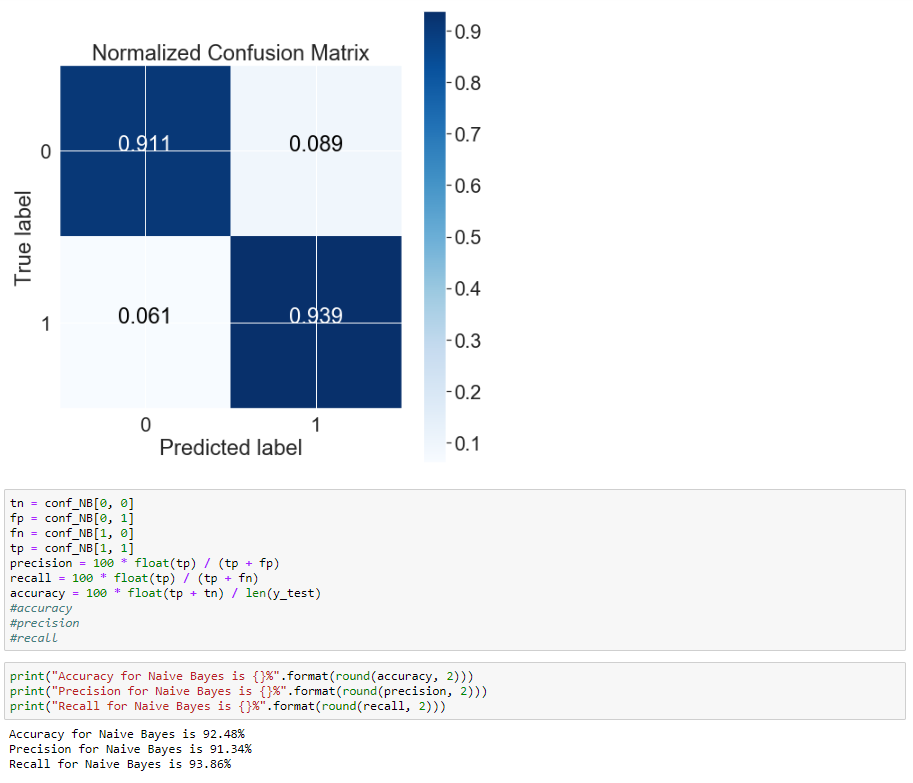


1. Classification Evaluation

We explore various classifiers from Scikit-Learn library and we look at their Precision, Recall, Accuracy score and run-time to ensure a both efficient and accurate classification. For each approach, we divide corpus into training and test dataset, select optimal parameters and create a confusion matrix illustrated as below. We also give a brief overview for each approach and please refer to our technical review for more details.

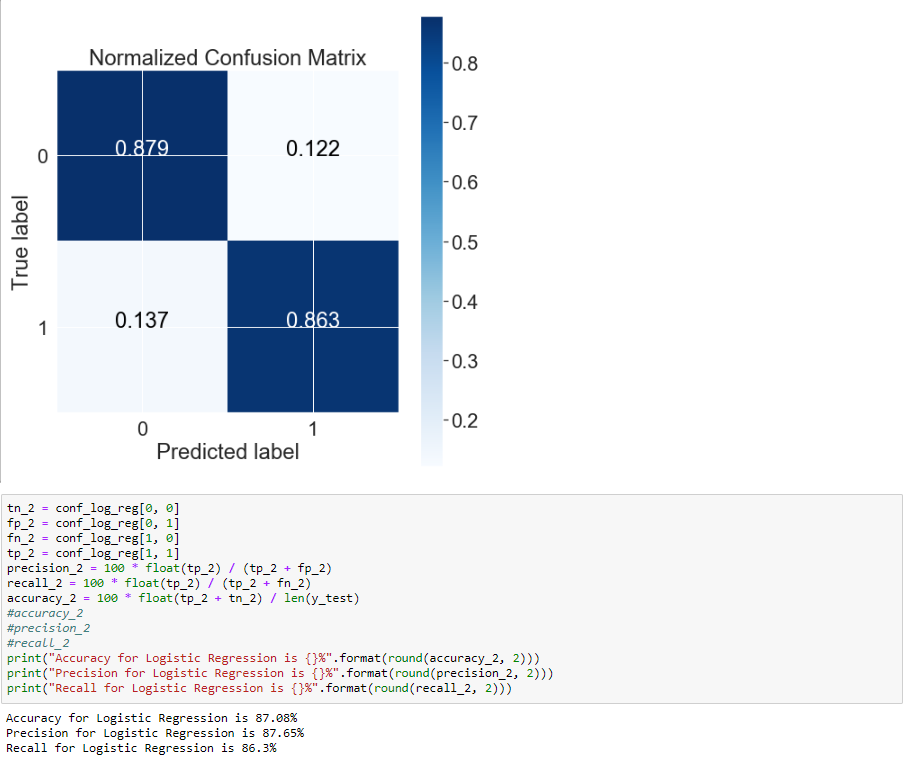
* Naïve Bayes

A Naïve Bayes classifier is a family of probabilistic algorithms, which uses Bayes' theorem in the classifier's decision rule, with an independent assumption between features.



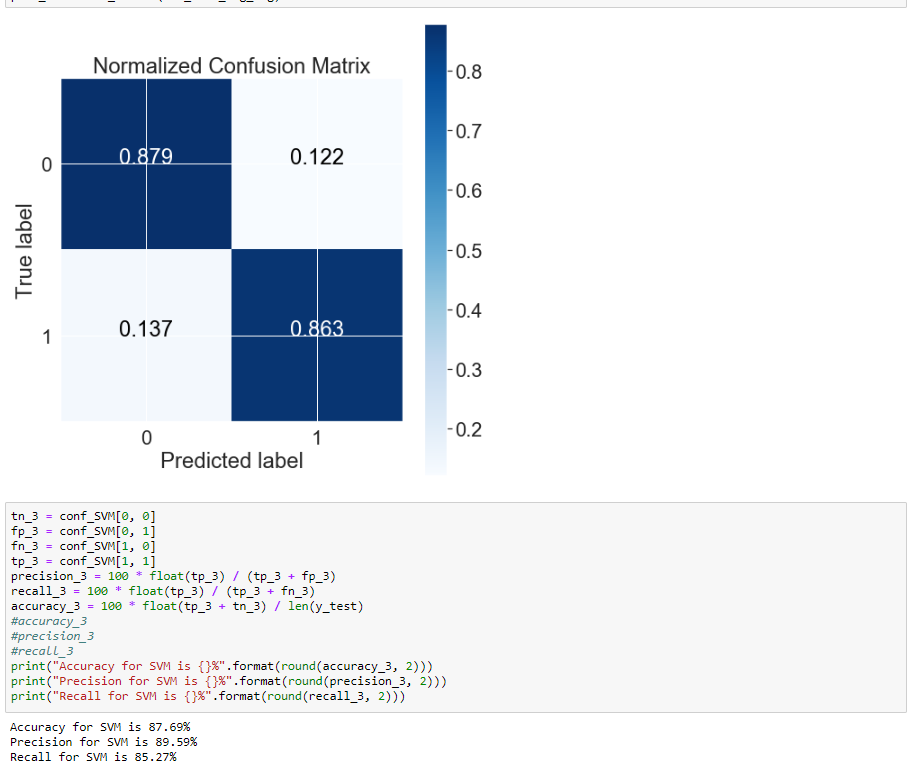
* Logistic Regression

Logistic Regression is also common to solve Binary Classification problem. The goal of Binary Classification is thus to find a model that can best predict the probability of a discrete outcome (notated as 1 or 0, for the “positive” or “negative” classes), based on a set of explanatory input features related to that outcome.



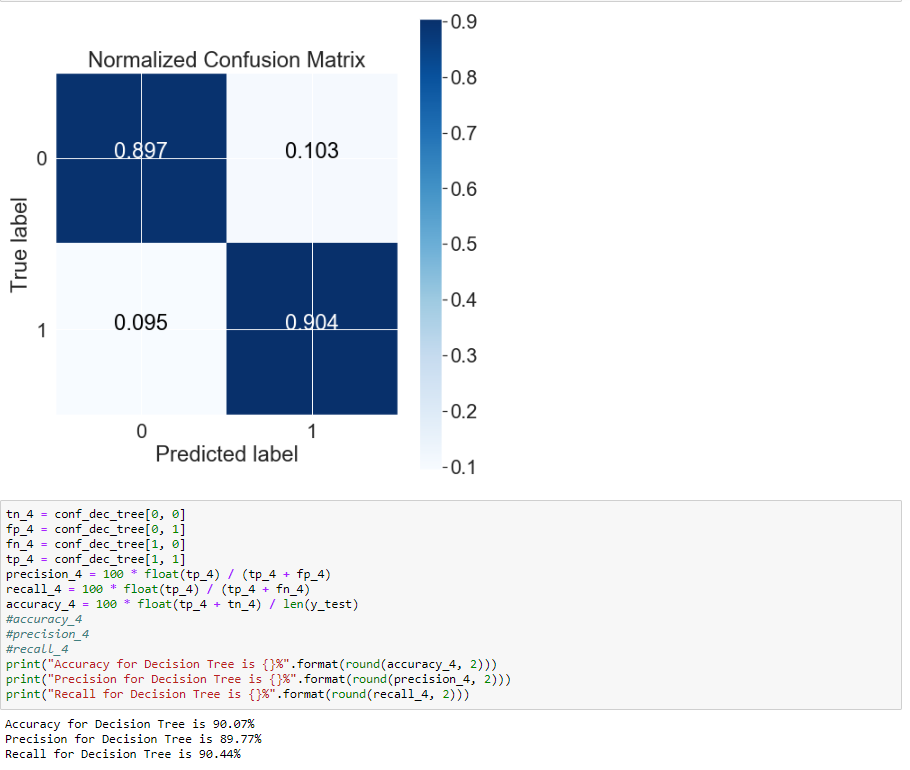
* Support Vector Machine (SVM)

A SVM is a classifier which uses annotated data for training to construct an optimal separating hyperplane/line in a multi-dimensional space which can be used to categorize new samples data into different groups. It is one of the key machine learning methods widely used for Sentiment Analysis.

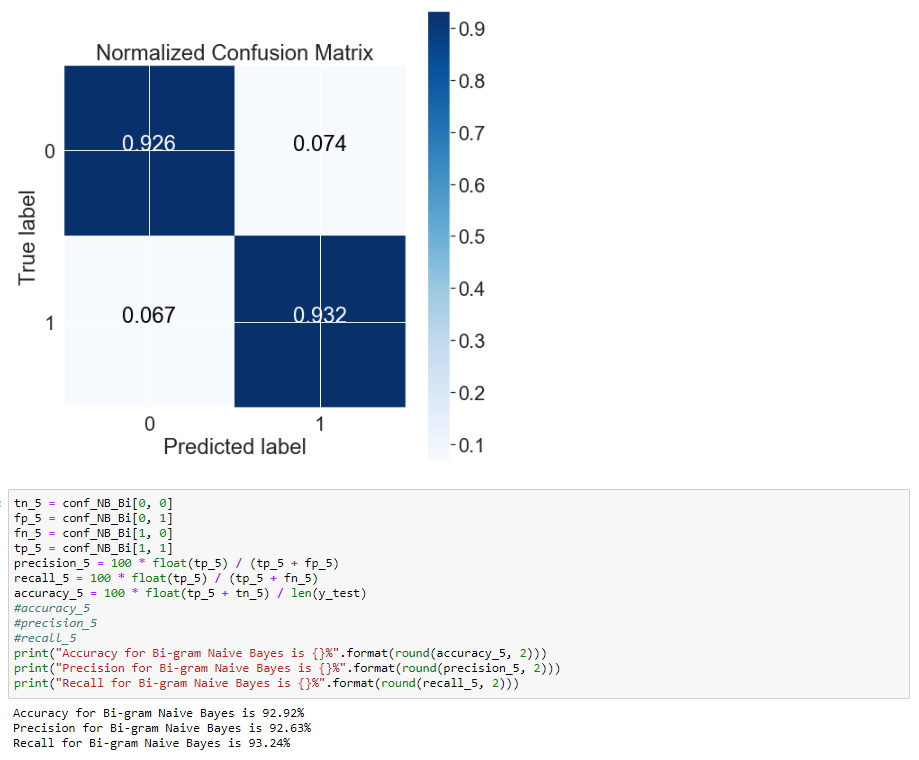


* Decision trees

Decision tree is a non-parametric learning method that predicts the value of a target variable by learning decision rules. Tree models where the target variable can take a discreate set of values are called classification trees and leaves represent class labels and breaches represent connections of features that lead to those labels. One of the advantages of decision trees is to learn inherent rules available in the dataset that are not available to the user.

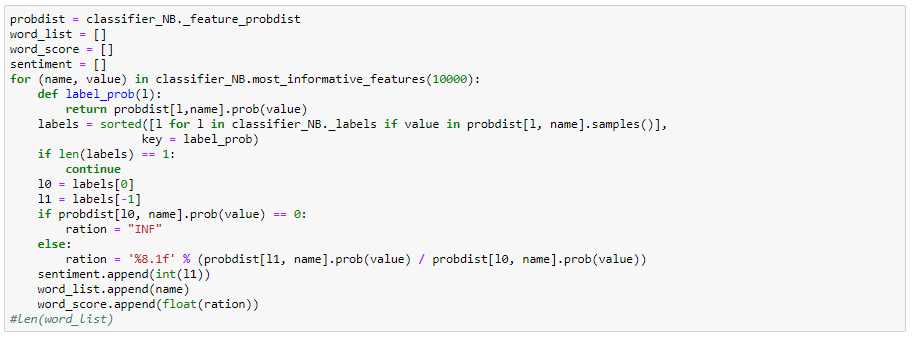


Naïve Bayes classifier has proved again that the simplest solutions are usually the most powerful ones. Despite other more advanced techniques, Naïve Bayes has achieved the highest accuracy score and reasonable run-time. Our next decision is how to continually improve Naïve Bayes classifier. In order to do so, we test bi-gram classifier and experimental results suggest that bigrams can substantially raise the performance of classifier especially for Recall.



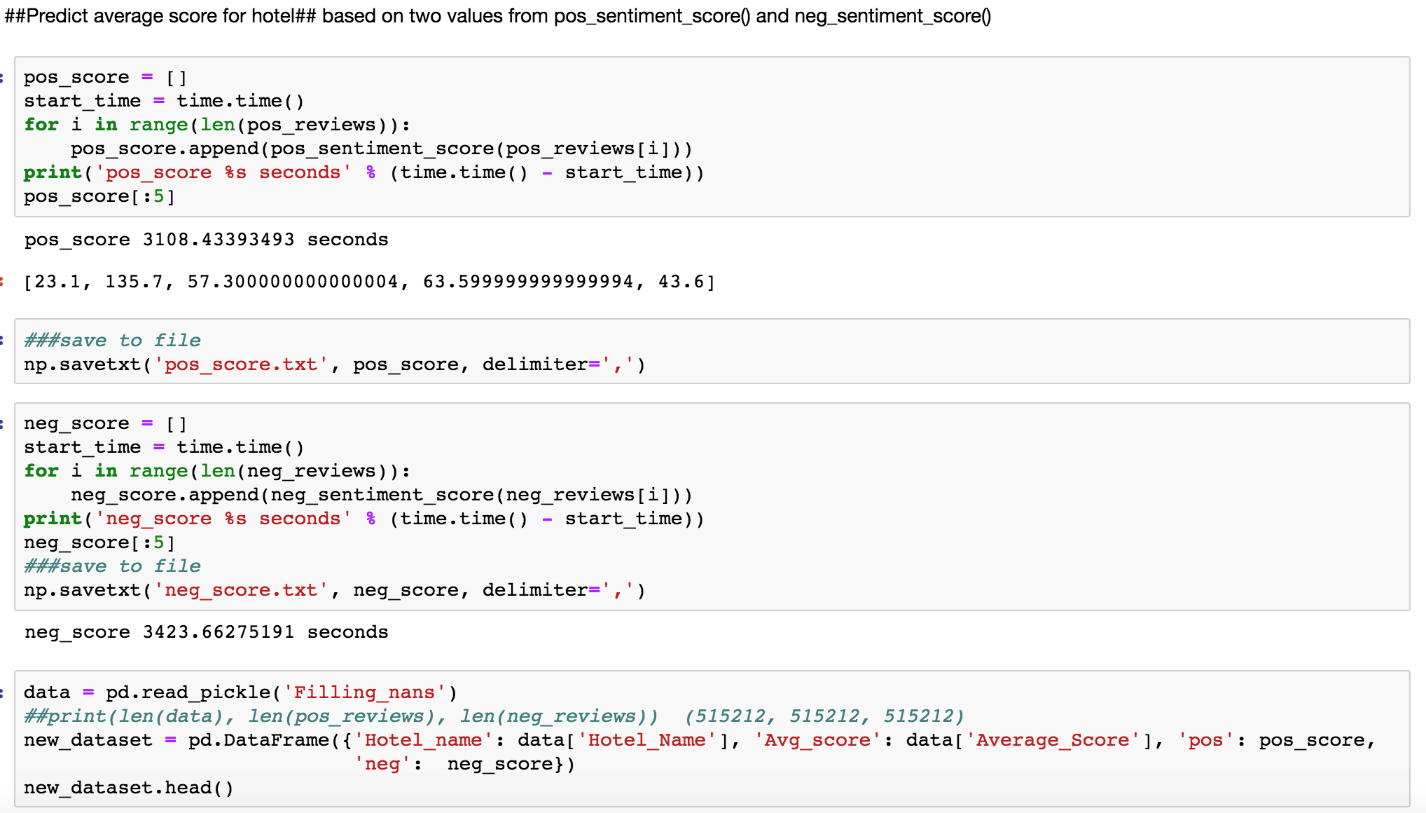
# Sentiment Analysis (aspect\_analysis\_data\_process.ipynb & aspect\_analysis.ipynb)

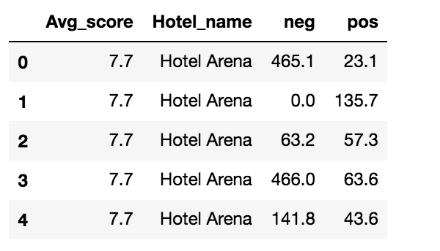
We switch to NLTK for further Sentiment Analysis, because 1) NLTK comes with all ideal built-in functions we need for Sentiment Analysis (textual tokenization, parsing, classification, stemming, tagging, semantic reasoning etc.) and 2) all of NLTK classifiers work with “featstructs”, which is a simple dictionary mapping a feature name to a feature value. It allows us to use simplified bag of words model where every existing word is a feature name with a value of True. From previous step, we have known that Naïve Bayes performs best with our dataset, so we first create our own classification model using Naïve Bayes classifier again. 70% reviews are used as training data for the classifier and the remaining 30% reviews are used to test and calculate accuracy score. Contrast to the previous classification where we use all the words as features, here we only use 10,000 most informative words for positive reviews and negative reviews. Save each word and its corresponding sentiment score to “pos\_word\_sen\_score.csv” and “neg\_word\_sen\_score.csv” for future use. Now we’re ready to put this preparation into use for three applications.





In order to compare each hotel based on various aspects, our project calculates each positive sentiment score and negative sentiment score based on above method for each positive and negative review (see details in “data\_process.ipynb”). Then save each score to “pos\_score.txt” and “neg\_score.txt”. This step may take longer time.





## Application 1: Strength of Sentiment in Reviews

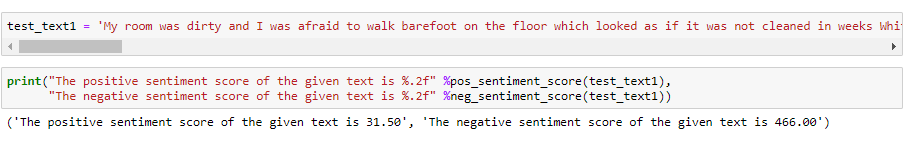
This application will estimate the strength of positive and negative sentiment in reviews. It takes a review as input and return a positive sentiment strength (ranging from 0) and a negative sentiment strength (ranging from 0) by calling pos\_sentiment\_score() and neg\_sentiment\_score() respectively. In this way, hotels will better understand customers’ experience by extracting their emotional tone from the reviews they post and use this insight to improve their business and gain competitive advantages. See details in “data\_process.ipynb”.



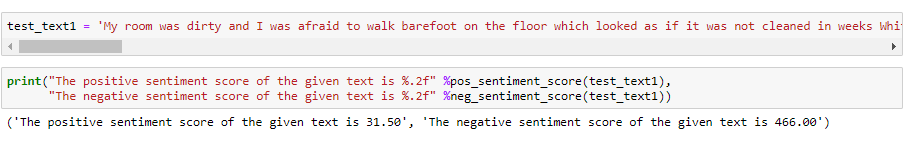
Test Case (test\_case.ipynb):

Input:

My room was dirty and I was afraid to walk barefoot on the floor which looked as if it was not cleaned in weeks White furniture which looked nice in pictures was dirty too and the door looked like it was attacked by an angry dog My shower drain was clogged and the staff did not respond to my request to clean it On a day with heavy rainfall a pretty common occurrence in Amsterdam the roof in my room was leaking luckily not on the bed you could also see signs of earlier water damage I also saw insects running on the floor Overall the second floor of the property looked dirty and badly kept On top of all of this a repairman who came to fix something in a room next door at midnight was very noisy as were many of the guests I understand the challenges of running a hotel in an old building but this negligence is inconsistent with prices demanded by the hotel On the last night after I complained about water damage the night shift manager offered to move me to a different room but that offer came pretty late around midnight when I was already in bed and ready to sleep.



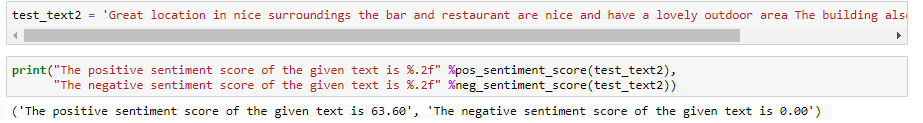
Output:



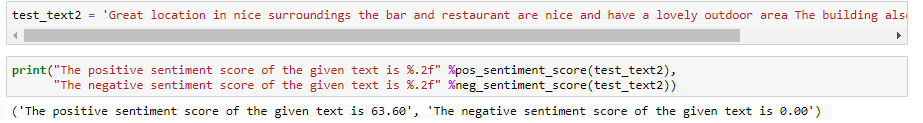
The given test review is classified as a **negative review** since the neg\_sentiment\_score is much larger than pos\_sentiment\_score. And since this review is coped from column "Negative\_Review", the result matches the right categorization.

Input:

Great location in nice surroundings the bar and restaurant are nice and have a lovely outdoor area The building also has quite some character



Output:



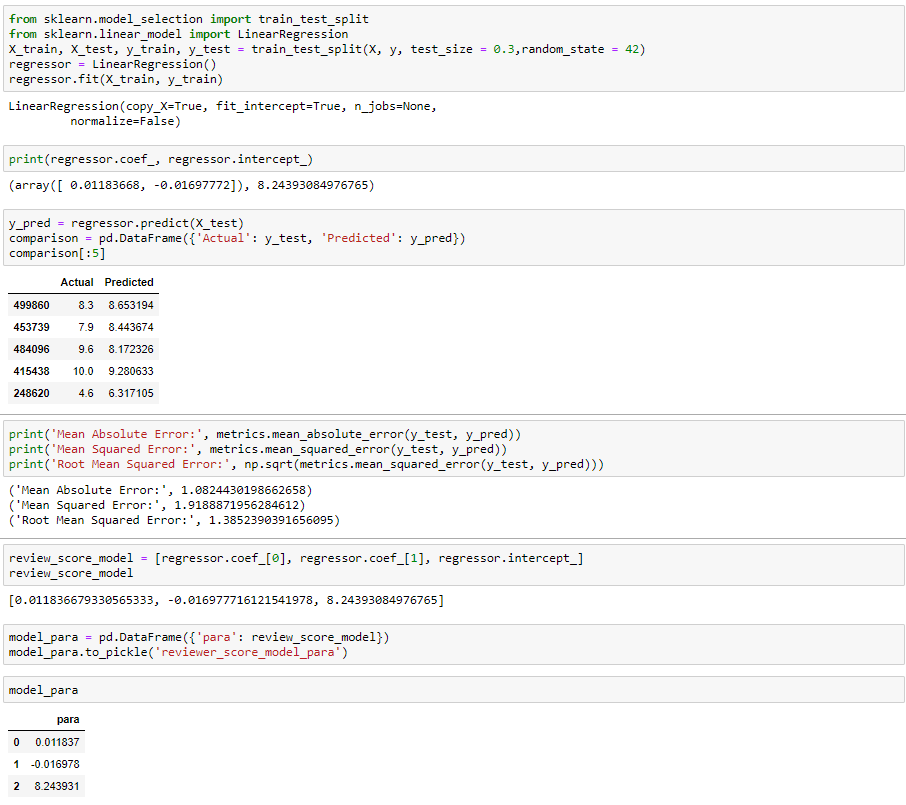
The given test review text is classified as a **100% positive review** since the neg\_sentiment\_score is 0. And since this review is copied form column "Positive\_Review", the result matches its right categorization.

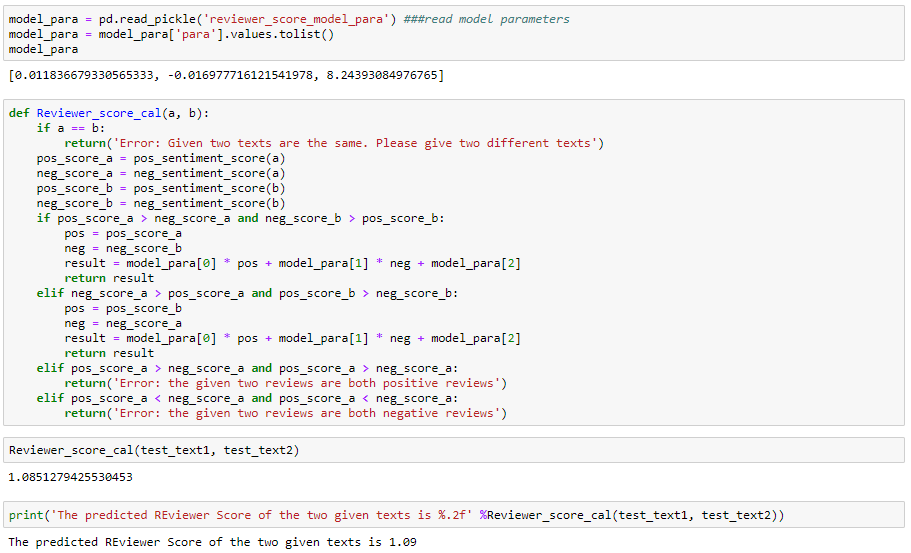
The two returned scores tell us how negative or positive a given review is, in other words, how dissatisfied or satisfied a customer is. The two test cases show that our tool works as desired.

## Application 2: Reviewer Score Prediction

As an extension of application 1, application 2 takes two reviews (no need to specify which one is positive and which one is negative, but they need to be different) as input and predicts Reviewer\_Score using the correlation between Reviewer\_Score and strengths of positive and negative sentiment (from application 1) in reviews by calling Reviewer\_score\_cal(). This application can be beneficial to hotels in some scenarios, especially where customers forget leaving an overall score. With our tool at their disposal, it’s easy and quick for hotel managers to generate a numeric feedback out of mixed reviews.

We use built-in linear regression model from Scikit-Learn library and Mean Absolute Error, Mean Squared Error, Root Mean Squared Error to evaluate its performance. See model details in “reviewer\_score.ipynb”.



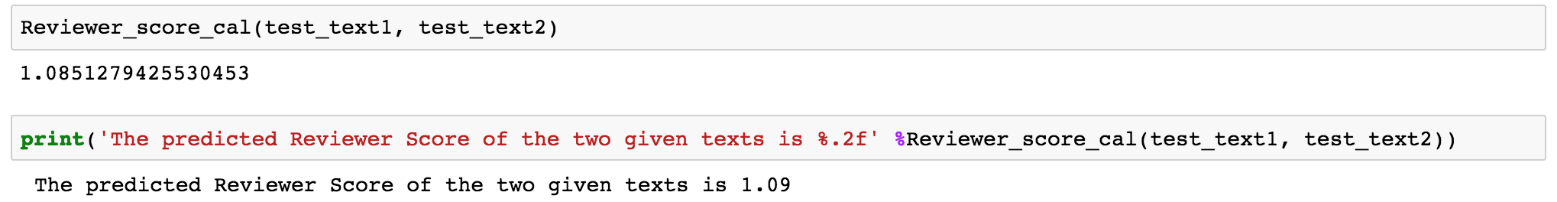


Test Case (test\_case.ipynb):

Input:

We continue to use two reviews from Application 1.

Output:



If these two reviews are given by the same reviewer to a certain hotel, we could predict the Reviewer Score for this hotel is 1.09 on the scale of 1 to 10 based on this experience.

## Application 3: Hotel Ranking List Based on Aspects

There are three different levels of Sentiment Analysis: Document-based, Sentence-based and Aspect-based. Aspect-based Sentiment Analysis has gained increasing popularity in both industry and academia, but only recently it has entered the domain of tourism. Please refer to our technology review for a detailed review of existing research and trendy applications.

In our work, we implement an Aspect-based opinion miner for hotel reviews and find important aspects for positive and negative reviews, based on which we rank all hotels by Average\_Score and Aspect-based sentiment analysis on reviews. First, we collect top 20 common words in positive reviews and negative reviews and save them in two .txt files called “freq\_pos\_20.txt” and “freq\_neg\_20.txt” respectively. By deeply looking at the 20 words from each file, the top 5 aspects are "staff", "location", "room", "breakfast", "bed" for positive and "room", "breakfast", "staff", "bed", "bathroom" for negative.

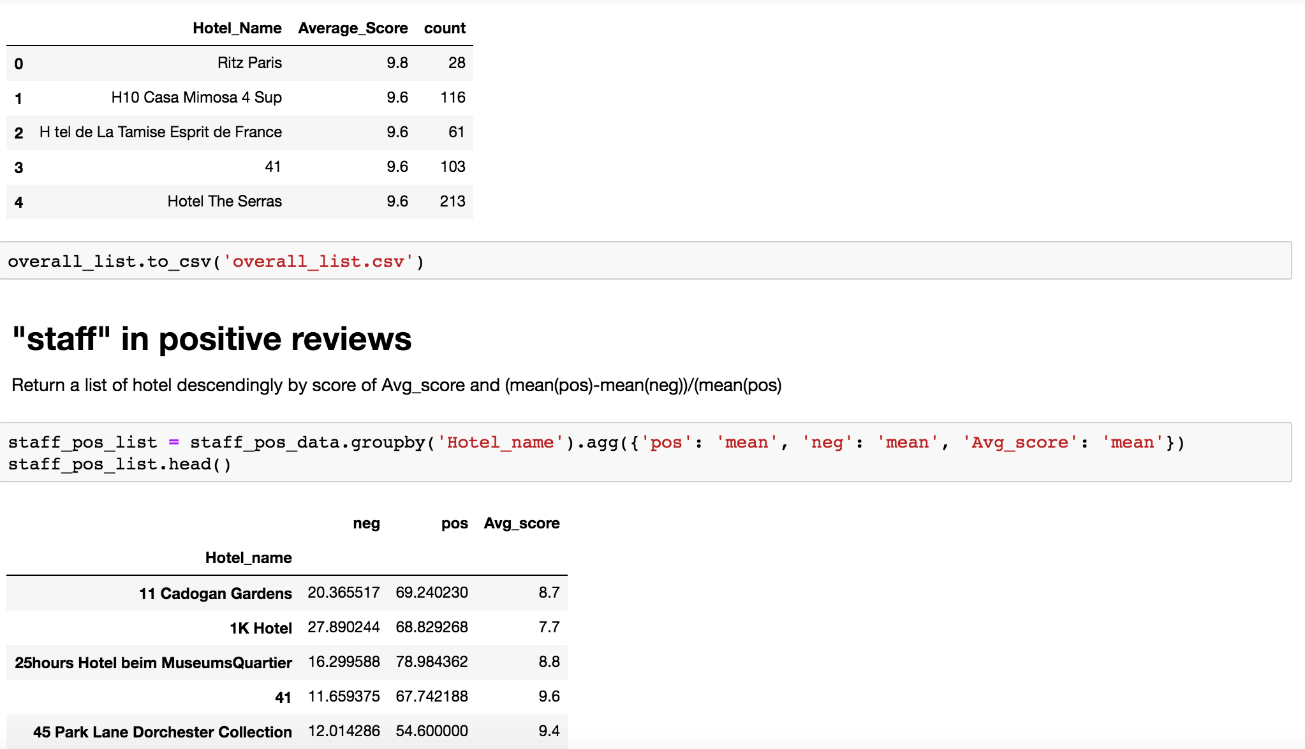
First apply the previous calculated positive sentiment score and negative sentiment score and group all the sentiment scores and Average\_Score by “Hotel\_Name”. Then save each aspect related data to pickle file.





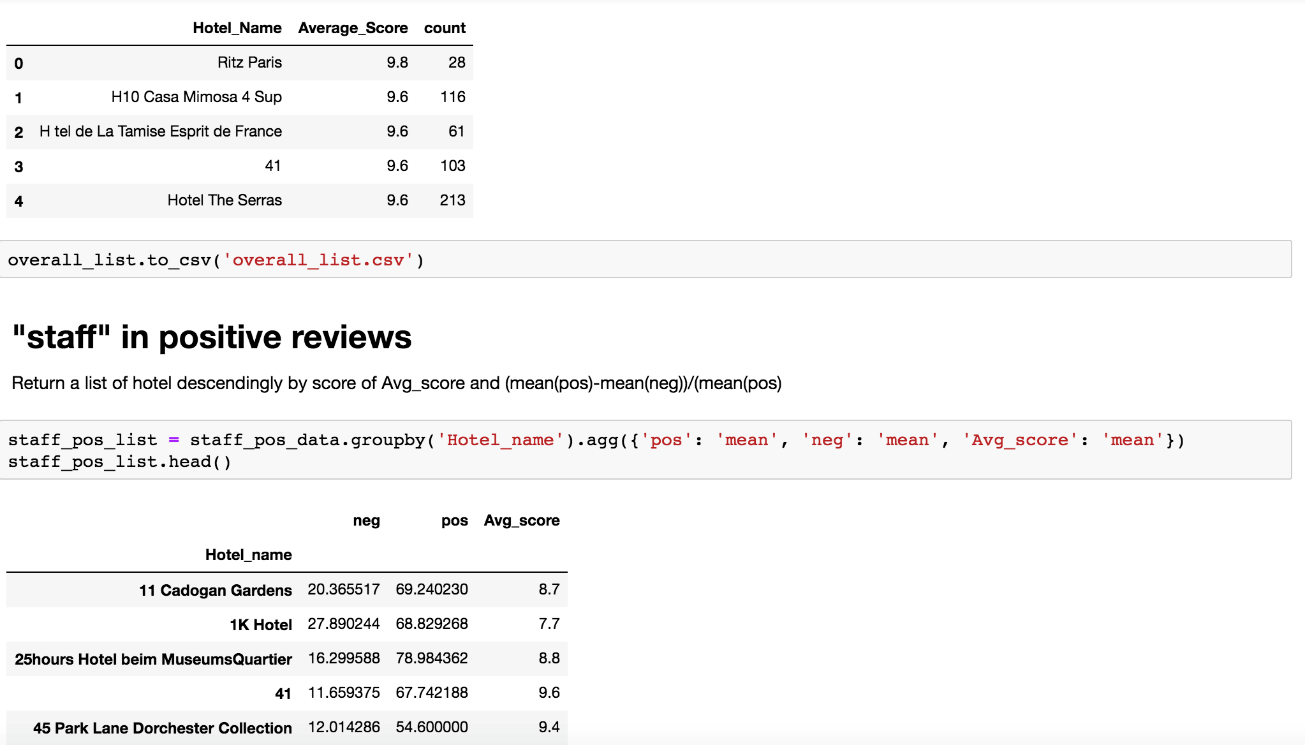
If a customer has no interest in any optional aspects, the customer will get a ranking list only based on descending value of “Average\_Score” by reading “overall\_list.csv”.

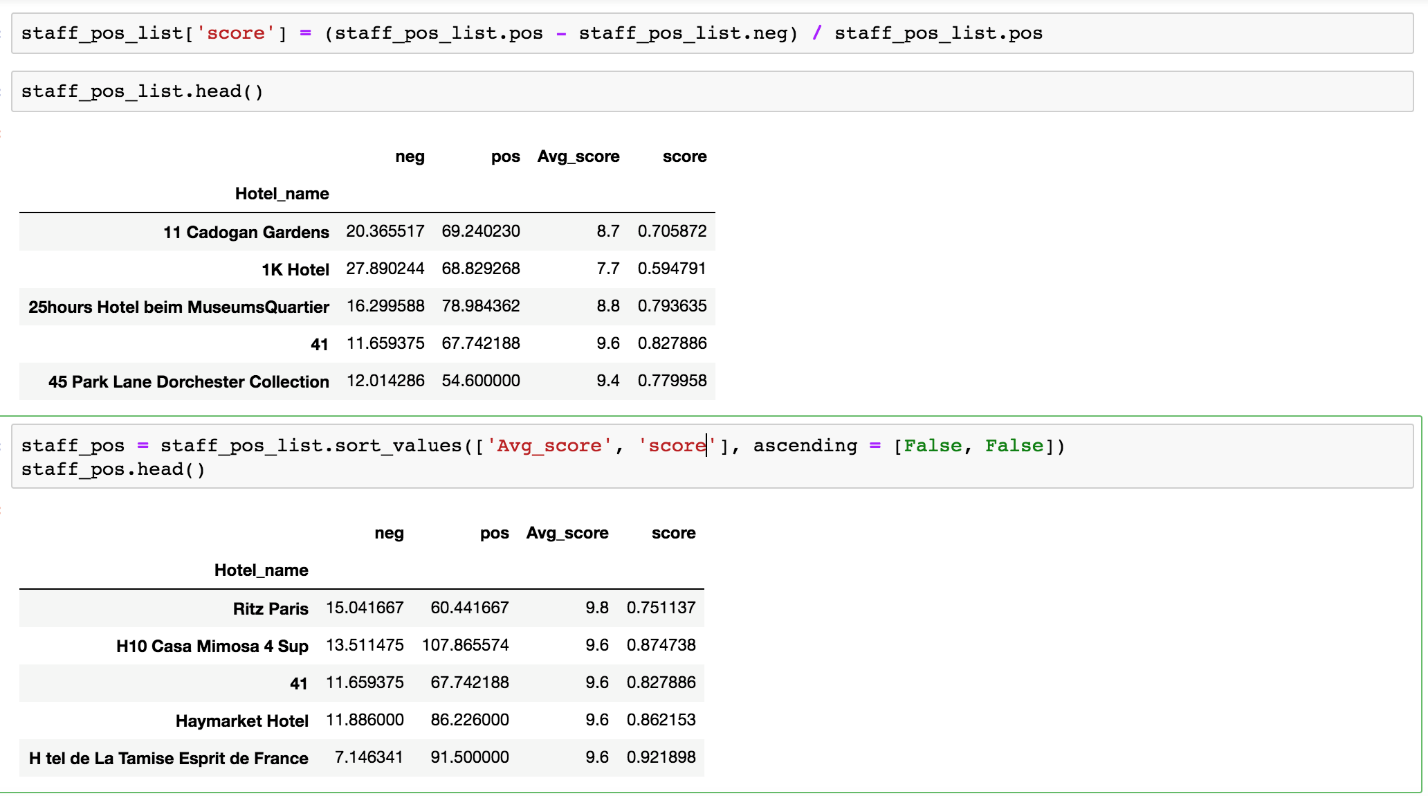




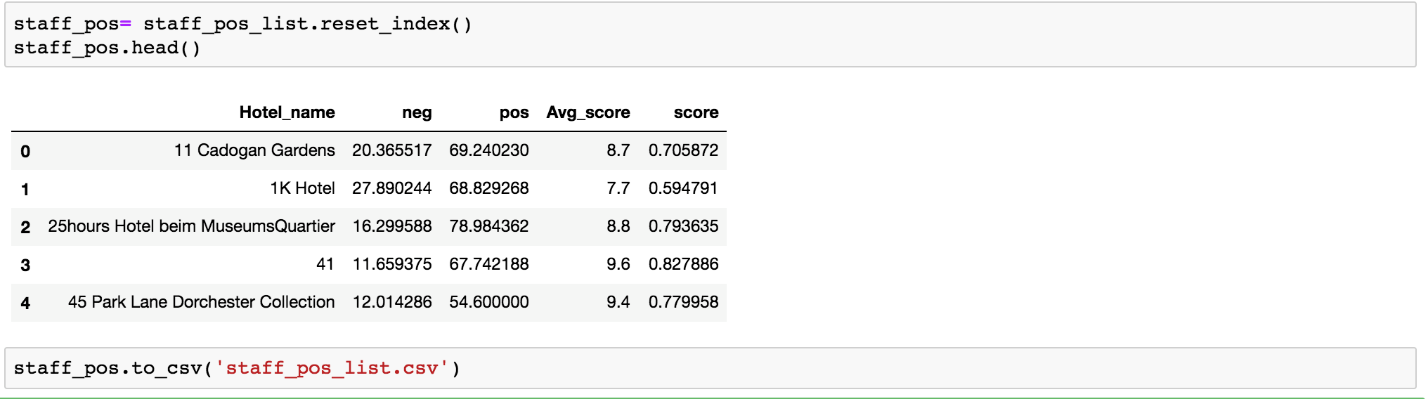
In order to compare hotel under each aspect, firstly read each aspect-related data from pick file then filter reviews based on aspect. Secondly, group positive sentiment score and negative sentiment score and Average\_Score by each Hotel\_Name and get mean values respectively. Ranking rules are as followings: if any two hotel has different Average\_Score, rank higher Average\_Score first; if two hotels have the same Average\_Score, compare mean ratio by the following formula, rank higher mean ratio first.

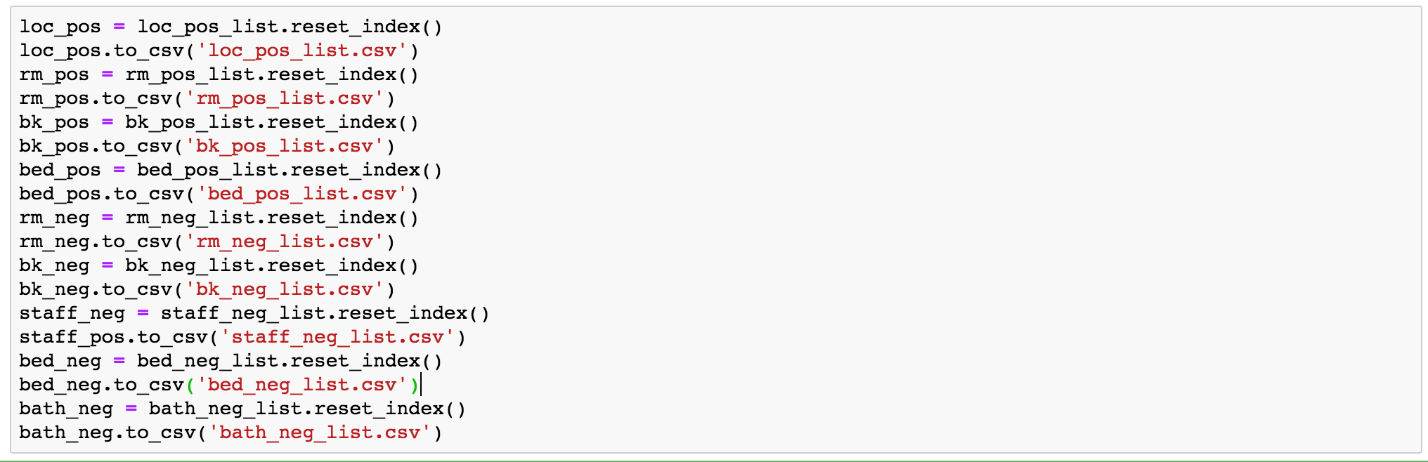






Each ranking list is saved to .csv file for user to read directly. 'loc\_pos\_list.csv' is a rankink list of hotels has "location" in positive reviews. 'staff\_pos\_list.csv' is a rankink list of hotels has "staff" in positive reviews. 'rm\_pos\_list.csv' is a rankink list of hotels has "room" in positive reviews. 'bk\_pos\_list.csv' is a rankink list of hotels has "location" in positive reviews. 'bed\_pos\_list.csv' is a rankink list of hotels has "bed" in positive reviews. 'rm\_neg\_list.csv' is a rankink list of hotels has "room" in negative reviews. 'bk\_neg\_list.csv' is a rankink list of hotels has "breakfast" in negative reviews. 'staff\_neg\_list.csv' is a rankink list of hotels has "staff" in negative reviews. 'bed\_neg\_list.csv' is a rankink list of hotels has "bed" in negative reviews. 'bath\_neg\_list.csv' is a rankink list of hotels has "bathroom" in negative reviews. Then visualize the result by google map via “folium” library.



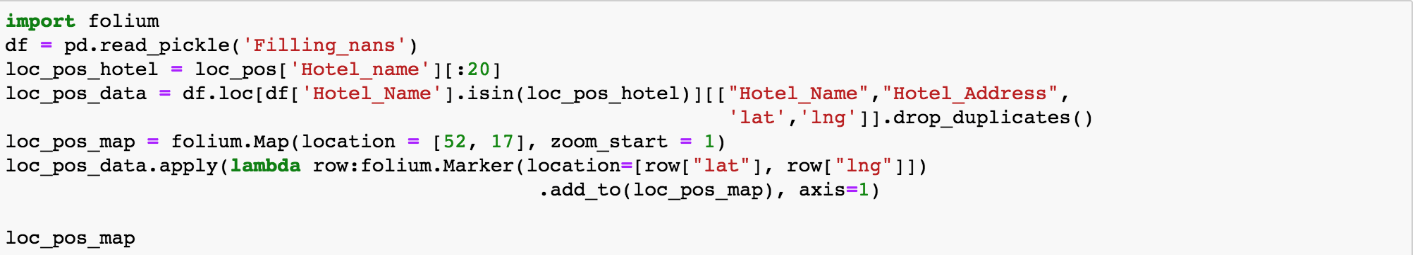


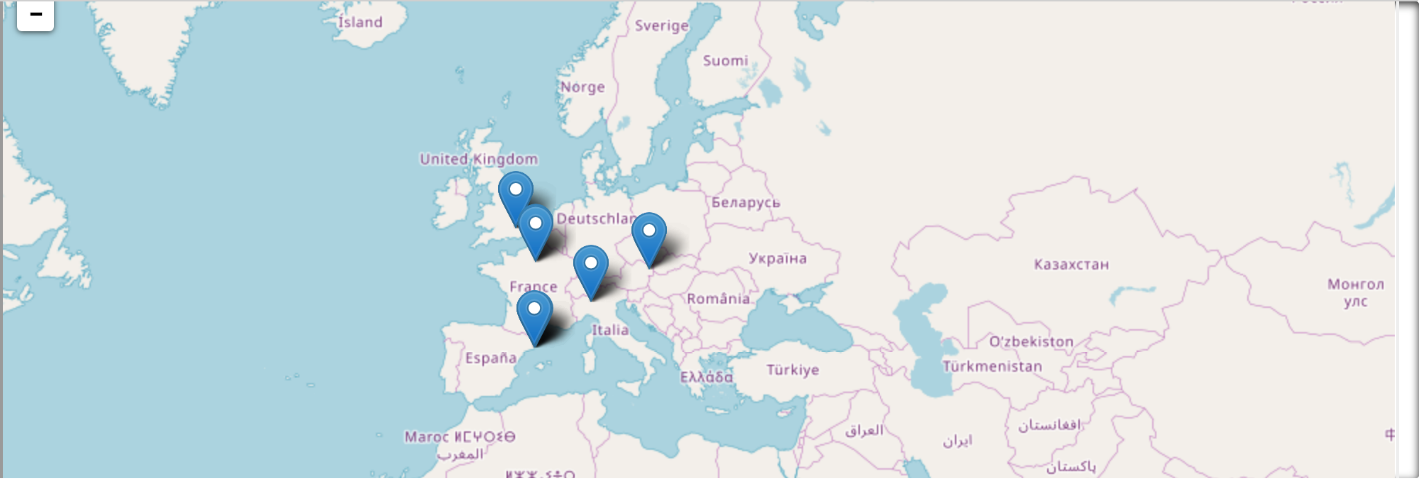
This way, hotel can address customers’ different needs which is critical to their experience satisfaction. For a potential guest who has high standards on location, this tool will return top 20 hotels with highest average score and best location-related reviews. Hotel Ranking List by aspect-based sentiment score is extremely helpful when it comes to hotel decision-making process. One question all the time is which hotel I should choose when there is a tie in their ratings. Our tool solves this problem by asking what matters most and rescues potential customers from a struggle to pick out useful comments from enormous amount of reviews.

Test Case (test\_case.ipynb):

Assume a customer is interested in “location” from positive reviews.







# Discussion

Tremendous amount of Hotel reviews are available on booking.com but it takes lots of time to read and analyze them, therefore for that purpose Sentiment Analysis is needed. The main contribution of our work can be summarized as below:

1. Based on relatively large amount of data

The corpus of 515k reviews is used to build and test review classifiers for Sentiment Analysis. It might take hours to run the whole script without cache.

1. Different classifiers and libraries are deployed and compared

We use different pre-processing strategies and machine learning approaches to determine the polarity of hotel reviews. In conclusion we determine that Bigram feature with Naïve Bayes classifier works best with our dataset.

1. Document-based Sentiment Analysis and Aspect-based Sentiment Analysis

By treating each positive/negative review as a document to be classified, our tool is able to associate customer feedback with overall sentiment scores telling hotels how happy or dissatisfied a customer is in general. To reach Sentiment Analysis’ full potential, our tool takes different aspects of hotel experience into consideration and provides how people’s sentiment varies when they are talking about room, staff, breakfast or location. There’re some good opportunities for hotels to re-evaluate areas of opportunity and growth and better understand their clients.

1. Useful for both hotel managers and potential customers

There is always room for improvement as hotels strive to give customer to best possible experience. Determining important features/aspects expressed in online hotel reviews is vital for hotel managers as mentioned in 3. In addition, potential guests using this tool are equipped with a more powerful weapon when booking their hotels. They are provided a chance to filter same-rated hotels by a feature that matters most and a ranking list based on the feature.

# Future Work

In the end, how might this tool be improved? The answer lies in the ability to derive additional insights from the data. Some directions of future work include:

1. Aspect-based Sentiment Analysis and Reviewer\_Score

Current work is being done to tie Reviewer\_Score with overall sentiment score using linear regression. Furthermore, prediction on Reviewer\_Score can be achieved though Aspect-based Sentiment Analysis and more advanced machine learning approaches. This will help answer questions like:

* How positive or negative opinions for each aspect contribute to overall sentiment score?
* What weighted combination of different aspects best predicts a Reviewer\_Score?

1. User Interface:

A user interface is the single most important element that plays a massive role in bringing in high volumes of users. If our tool’s user interface can evolve from a command line interface to a user-friendly graphical web browser, more people are encouraged to use it and provide more feedback on how to continually improve its functionality.

# Reference

Adeborna, E., & Siau, K. (2014). An Approach to Sentiment Analysis – The Case of Airline Quality Rating. *Pacific Asia Conference on Information Systems (PACIS) .*

Alaei, A., Becken, S., & Stantic, B. (2017). Sentiment analysis in tourism: Capitalising on Big Data. *Journal of Travel Research*.

Brob, J. (2013). *Aspect-oriented sentiment analysis of customer reviews using distant supervision techniques.* PhD Thesis, Department of Mathematics and Computer Science, University of Berlin.

Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Association for the Advancement of Artificial Intelligence*.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Synthesis Lectures on Human Language Technologies.

Marrese-Taylora, E., Velasqueza, J. D., Bravo-Marquezb, F., & Matsuoc, Y. (2013). Identifying Customer Preferences about Tourism Products using an Aspect-Based Opinion Mining Approach. *17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013*, 182 - 191.

PAurchana.P, PIyyappan.R, & PPeriyasamy.P. (2014). Sentiment Analysis in Tourism. *IJISET - International Journal of Innovative Science, Engineering & Technology*, Vol. 1 Issue 9.

Shi, H.-X., & Li, X.-J. (2011). A sentiment analysis model for hotel reviews based on supervised learning. *2011 International Conference on Machine Learning and Cybernetics.*

T, C. C., & Joseph, S. (2014). Aspect based Opinion Mining from Restaurant Reviews. *International Journal of Computer Applications*.

Zhai, C., & Massun, S. (2016). *Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining.* New York: Association for Computing Machinery and Morgan & Claypool.