DECEMBER 12, 2021

HOTEL REVIEW ANALYSIS

DATA ANALYTICS FINAL PROJECT

TEAM PANDAS LEARNING

Mary Kiangi (mdk2170), Hoa Duong (hhd2115), Joseph High (jph2185), Jiawen Li (jl6121), Ziyun Lu (zl2961)

1. Introduction

Reviews and ratings have revolutionized traditional word-of-mouth marketing into a viral form of feedback that can influence consumers' opinions. When we consider reviews and ratings, both businesses and consumers can benefit. Reviews help consumers voice their needs and act as a form of recommendation, a high-value social proof. Also, reviews help business owners to identify potential areas of improvement, measure overall customer satisfaction, and create micro-marketing campaigns[1]. With the constantly growing volume and variety of review data, effective analysis becomes of paramount importance.

Our project is based on reviews and rating analysis with a case study of hotels in the US. We seek to provide insights to both the customers and the hotel owners. We used data visualization and text mining techniques such as word clouds, NRC Sentiment Analysis, and Vader Analysis to explore the patterns and features from the dataset. Next, through feature engineering, we augmented roughly 28,000+ features from the raw review texts, which were later used as inputs to the predictive models. Finally, we built, tuned, and compared the performances of various machine learning models such as Logistic Regression, Decision Tree, Random Forest, and Bagging to categorize hotels based on the review texts.

2. Data Cleaning and Preprocessing

2.1 Data Cleaning

We concatenated three different datasets from Kaggle into a dataset of 55,912 entries as shown in Figure 1. This left us with information on the hotels, reviews, numeric rating, as well as information on the reviewers. As our analysis focus on the ratings and reviews, we dropped the observations with missing ratings or missing review texts. We converted missing review titles into empty strings and concatenated all review titles to review texts to create a review column.

Taking a deeper look at the review column, we noticed that some reviews placeholders and dummies, we removed these reviews. In addition, some reviews were not legible. They either contain non-ASCII characters or foreign-language texts. We used Google Translator to convert these reviews into English texts and removed the non-ASCII characters

2.2. Data Processing

The obvious reviews rating scale is 1 to 5. However, we noticed that there were some reviews with a rating of 0 or a rating greater than 5, with a maximum of 10, as shown in Figure 2. Though only 0.61% and 0.65% of total observations having ratings of 0 and ratings greater than 5, respectively, we did not remove these observations. We checked if there exists another rating scale of 0-to-10. If a hotel employed a 0-to-10 rating scale, a rating of 5 would not be considered a good rating, but rather a neutral rating. Removing ratings greater than 5, we would mistakenly label the not-so-good reviews as good reviews, thereby impacting our model performance. We found that there are 253 hotels with ratings of 0 and 5 hotels with ratings greater than 5. Among all observations, there are 9.6% and 1.4% of total observations associated with these 253 hotels and 5 hotels, respectively. As such, we rescaled these ratings from a 0-to-10 scale to a 1-to-5 scale. Figure 3 shows the rating distribution after all ratings were rescaled to the 1-to-5 scale.

We investigated the hotels' locations. As shown in Figure 4, there are hotels located outside of the US. In fact, 14.7% of total reviews are for hotels outside of the US. As we wanted to focus our analysis on hotels in the US, we excluded these reviews for the purpose of this analysis. The result is shown in Figure 5. From the coordinates of the hotels, we also extracted the hotels' state using reverse geocoding and created a heatmap, as shown in Figure 6. It can be seen from Figure 5 and Figure 6 that the distribution of hotels' locations is reasonable given the geographical location of the US.

3. Exploratory Data Analysis

3.1 Rating Analysis

The first question we wanted to answer was "What are the best and worst hotels in the country?" As we have about 3,000 hotels with around 55,000 reviews, we grouped the reviews by hotel name and plotted the average ratings by the hotel in Figure 7. We can see that there are a lot of hotels with an average rating of 5, an extreme. Many, hotels with an average rating of 1 or 5 only have one review each. As such, the "average" ratings are in fact calculated from a single review. This aligns well with what we would expect from travelers: one would leave a review if he or she is particularly dissatisfied or recommend if satisfied with the service. To account for this, in Figure 8, we plotted average ratings by the hotel, including only hotels with more than 5 reviews.[2] Not surprisingly, the number of hotels with an average rating of 1 or 5 decreased significantly. In contrast, most hotels had an average rating ranging from 3.75 to 4.5. Figure 9, in which we plotted the hotel reviews by rating categories, showing that the "Very good" reviews, reviews whose ratings round to 5, are the most common ones, followed by the "Moderately good" reviews, those whose ratings round to 4. For this, we also explored an alternative, where we considered only three categories instead of five, as shown in Figure 10.

[1] Positive online business reviews are worth a great deal https://www.e-satisfaction.com/7-reasons-why-customer-reviews-are-important/

[2] All hotels include those with 1- 4 reviews are included in all analysis

Broken down by month, we can see seasonality in traveling, with a spike in the summer months, especially in July, as seen in Figure 11. Further broken down by month and rating category shown in Figure 12, we see the trend once again that travelers tend to leave more extreme reviews, with the number of "Very bad" and "Very good" reviews increasing in the summer months as well. This makes sense, considering that people tend to travel more in the summer thanks to school breaks, and that weather is generally warmer and nicer in the summer, leading to more hotel usage. Hotel owners can focus their resources on popular seasons to increase their profits.

3.2 Review Analysis

3.2.1. Word clouds

Other than seasonality, to increase their profits, hotel owners may also be interested in knowing what their customers care about the most in their properties. After removing the common words like "hotels," "stays," and "room," as well as the short words like "I" and "u," we generated word clouds from all reviews. Looking at the word cloud in Figure 13, we can see that many reviews mention "clean" and "staff," along with "friendly" and "comfortable," implying that travelers' tend to pay more attention to the cleanliness of the property, the friendliness of the staff, and the comfort of their overall experience. We verified this by looking closely at the word clouds for reviews with ratings of 5 in comparison to reviews with ratings of 1. As shown in Figure 14a and Figure 15b, we can see that the good reviews mentioned a lot of "clean" and "staff." There are also a lot of mentions of "friendly," "helpful," "comfortable," "convenient," "location," which all hint that the good hotels are the ones that are clean and comfortable, with friendly and helpful staff, and at a convenient location. On the other hand, the bad reviews are the ones about hotels that are "dirty," "booked," or even "smell." With this information, hotel owners could tailor their hotels' features and focus on keeping their hotels clean and training their staff to gain a better reputation.

Taking the word clouds one step further, we considered reviews for the best and worst hotels, as indicated by the average rating by hotels discussed above. To ensure we have long enough texts for each hotel, we considered the top three and bottom three hotels with more than 5 reviews. Figure 16 tells a similar story as far as customers' attention. However, we can see that the Litchfield Inn, while being one of the worst hotels, had many good words, such as "great," "location," "beach," and "oceanfront." This emphasizes a drawback of word clouds, being that they only show words and not the context of the words. Thus, we wanted to examine reviews summarization to better understand what the customers care about.

3.2.2. Review summarization

As we see in the word clouds, review ratings may not relate to the actual review context. A low rating may be given to a hotel with positive reviews. This problem may be collocated with the problem of advertising, where the hotel will put up their best description and imaging for their properties while the reality is otherwise. We decided to do review summarization to enable customers to get opinions from peers who have had first-hand experience. We first put together all hotel reviews for each hotel. Then using gensim.summarization, built a function "get_summarization" which if given a hotel name and number of word count will summarize reviews from the hotel as illustrated in Figure 17. This way the feedback is more accurate.

3.2.3. Aspect Classification with LDA

We looked to understand the focus of the reviews to get a sense of what people are talking about. One way could be to read through each review, which is impossible given the number of reviews. Another way would be to use review summarization as shown above, but with increasing data, it becomes difficult to keep track of what you are looking for. We decided to do topic modeling on the reviews using Latent Dirichlet Allocation (LDA). We extracted the review text and title data and used LDA to classify the reviews into topics. With several trials for two to eight topics, we picked three topics as the optimal number of topics. We visualized these topics using word cloud and pyLDAvis tool as shown in Figure 18 and Figure 19 to enable us to draw powerful insights from the results. We see the first topic focused on service where great importance is given to "staff," "time," "service," and "appreciation"; this reflects on customer service provided by the hotels. Another topic is centered around "room," "bed," "bathroom," and "water," all of which speak to the property itself. Finally, we see a focus on "location," "area," "convenient, "place," and "comfortable," which relate to the frequent travelers such as professionals and bring an emphasis on location and experience. These results can be used to provide actionable insights for hotel owners.

3.2.4 Vader Analysis

Vader sentiment analyzer was applied to the review texts and titles in this dataset. Vader is a rule- and lexicon-based framework for sentiment analysis, with support for intensity estimation. Each review and review title can be classified as having an overall negative, neutral, or positive sentiment polarity based on its compound sentiment score. This produces 4 Vader sentiment score features. We perform the analysis in 3 columns, text, title, and the text and title combined. We obtained 3 compound scores, named review polarity, title polarity, and review and title polarity.

We grouped the dataset by hotel name, counted the number of rating observations, and averaged the compound scores across different observations. We categorized the hotels as either Terrible, Bad, Neutral, Good, or Great based on the following threshold values.

Terrible	review polarity < -0.4
Bad	-0.4 <= review polarity < -0.1
Neutral	-0.1 <= review polarity < 0.1
Good	0.1 <= review polarity < 0.5
Great	review polarity >= 0.5

We plotted the distribution of hotel review categories, where it is heavily skewed towards Great and Good. Therefore, we performed sentiment analysis on a selected part of our dataset, where we focused on the hotels in the category of Good and with more than 10 reviews. We defined popularity as a hotel's average rating multiplied by the total number of reviews and mapped the 10 most popular hotels.

4. Feature Engineering

To extract more features about the reviews and integrate those features into our predictive models, we applied text mining techniques on the translated, preprocessed review texts. Reviews convey important information about the hotel location, customer service quality, room service quality, and other reasons why a certain hotel is rated as Great or Terrible.

From the texts, we were able to generate two kinds of features: meta-features and text features. Meta features are related to only the structure of the review texts. Intuitively, a long review with a rich vocabulary provides more details and insights than a short review with a small vocabulary.

Text features are concerned about the content of the reviews. Through Vader Analysis, we calculated statistics on how positive, negative, and neutral a review is and combined these results to give a compound sentiment (higher = more positive) for the review. In addition, emotion analysis attributes 8 emotional scores based on NRC data. Finally, the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer converted a collection of raw review texts to a matrix of TF-IDF features -- the overall document weightage of every single word in the review text corpus.

Meta Features (structure of the text)	**Text Features (the content of text)**
• Sentence length (characters & words) • Word length • Percentage of unique words • Stopword count	• Vader Analysis: positivity/ negativity • NRC Emotions: Scores for 8 emotions • TF-IDF: TF-IDF Vectorizer

Through the transformations mentioned above, we were able to augment the data set with 28,000+ additional features related to the review text itself.

5. Model Development and Outcomes Analysis

During the model development process, four different learning problems were explored. Namely, regression, five-class classification, three-class classification, and binary classification. For each approach, four classification algorithms were evaluated: decision tree, random forest, bagging classifier, and logistic regression. Model features were developed based on term frequency inverse document frequency (TF-IDF). Furthermore, the data for each set of models was randomly partitioned into an 80:20 train-test split.

As previously discussed, there is a severe class imbalance with respect to the ratings whereby a large majority of the population corresponds to ratings greater than or equal to 4. To treat this, the data set was augmented by oversampling the minority rating classes using an approach referred to as Synthetic Minority Oversampling Technique (SMOTE). This technique does just what the name implies, oversamples the minority classes by generating new observations from the existing ones. SMOTE was run on the training set for each set of models. Model performance before and after data augmentation using SMOTE was evaluated. In each case, model performance declined on the oversampled dataset, though only marginally. In the following subsections, we summarize and opine on the performance of each set of models. Please note, only the performance after data augmentation was conducted is discussed; please refer to the attached Python notebook for a more granular overview of model development.

5.1 Regression

Several regression techniques were developed on a combination of both review text and review title text, where the dependent variable was defined as the continuous rating in the raw dataset.

In all, six regression models were developed, with the support vector machine model exhibiting the strongest performance, as measured by R^2 and mean-square error (MSE). Results are displayed in Table 1. However, with a R^2 of 0.572 and an MSE of 0.704, the SVM model is still considered relatively weak performer.

5.2 Classification

Based on the term frequencies, as illustrated in the word cloud plots in Figure 14b, it appears that for reviews with ratings above 2.5, many of the most frequently used words are similar, like clean, good, and staff. On the other hand, reviews with ratings below 2.5 do not appear to have many frequently used words in common.

To analyze this further, a Vader analysis was performed on the review text. The resulting compound polarity scores were compared to the corresponding rating. The results indicate that the correlation between a review's rating and the compound score of the corresponding review text is quite low. There is no clear relationship between the two. This suggests that performance of classifiers with five classes may be low. This is demonstrated in the following subsection where we summarize the results for five-class classification.

5.2.1 Five-Class Classification

An additional column was added to the working data frame which grouped the ratings into one of five categories. In particular, the ratings from the raw dataset were rounded to the nearest integer. Four different classification models were subsequently trained on the TF-IDF features with the new column set as the dependent variable. The accuracy of each model is displayed in Table 2.

Based on these results, the random forest model exhibits the strongest performance, as measured by accuracy. However, an accuracy score of 0.52 isn't generally considered strong performance.

5.2.2 Three-Class Classification

For three-class classification, the ratings were partitioned into three groups: Bad, Neutral, and Good which were encoded by 0, 1, and 2, respectively. An additional column was added to the dataset which grouped the ratings in this way. Based on the word clouds displayed in Figure 18 there appears to be a larger degree of separation between the most frequently used words among the three categories. This suggests that 3-class classification models trained on this data using a dependent variable with these 3 classes may perform more strongly than classifiers on 5 classes. As such, the same four classification techniques were used to train four classifiers on the review text data and the new target variable with three classes. The accuracy of each model is displayed in Table 3.

As expected, model performance is much stronger for three classes. In this case, logistic regression exhibits the strongest performance while performance of the other classifiers is also strong. Despite the increase in performance, we were interested if we could do any better, so we generalized the reviews even further, into two classes.

5.2.3 Binary Classification

Four binary classifiers were trained on review text such that ratings greater than or equal to 3 were considered good and everything else was considered bad (or denoted by a zero). Binary classification performance was much stronger, which is no surprise as there is a large degree in term frequency between ratings above 3 and those below 3, as demonstrated by the term frequencies displayed in the word cloud plots in the Appendix. Performance of each model is displayed in Table 4.

On a binary target, the logistic regression and random forest models outperform the decision tree and bagging classifiers, as measured by accuracy, precision, recall, and the area under the ROC curve. The ROC curve and precision-recall plot are displayed in Figure 20 and Figure 22.

From Figure 24 and Figure 25, there appears to be a significant disparity between the compound polarity scores for the review text and the corresponding title text; binary classifiers were also trained on the title text. As evidenced by the results in Table 5 the classifiers trained on the title text features perform at a similar level to that of classifiers trained on the review text, with logistic regression again exhibiting the strongest performance.

6. Future Improvements

Firstly, we will try to generate review summarizations for hotel recommendations by considering similarities between reviewers. The goal is to build an app or extend functionalities of available apps to give user recommendations and real experience in a click.

Secondly, we could optimize the runtime for data-preprocessing and model training functions. For example, after feature engineering, we have too many features. We could run a feature selection pipeline to reduce overfitting, improve accuracy and minimize training time.

Finally, we are interested in exploring other predictive models such as Neural Networks, LSTMs, and more recently Transformers for sentiment analysis and topic detection.

References:

1. https://eric.clst.org/tech/usgeojson/
2. https://www.kaggle.com/datafiniti/hotel-reviews
3. https://freetext.ai/blog/review-analysis-for-product-reviews/

Appendix 1: Data Staging

Figure 1: Database Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55912 entries, 0 to 55911
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   address             55912 non-null  object
 1   categories          55912 non-null  object
 2   city                55912 non-null  object
 3   country             55912 non-null  object
 4   latitude            55826 non-null  float64
 5   longitude           55826 non-null  float64
 6   name                55912 non-null  object
 7   postalCode          55857 non-null  object
 8   province            55912 non-null  object
 9   reviews.date        55653 non-null  object
 10  reviews.rating      55050 non-null  float64
 11  reviews.text        55889 non-null  object
 12  reviews.title       54288 non-null  object
 13  reviews.userCity    30427 non-null  object
 14  reviews.username    55869 non-null  object
 15  reviews.userProvince 30221 non-null  object
dtypes: float64(3), object(13)
memory usage: 6.8+ MB
```

Figure 2: Rating Distribution on Raw Dataset

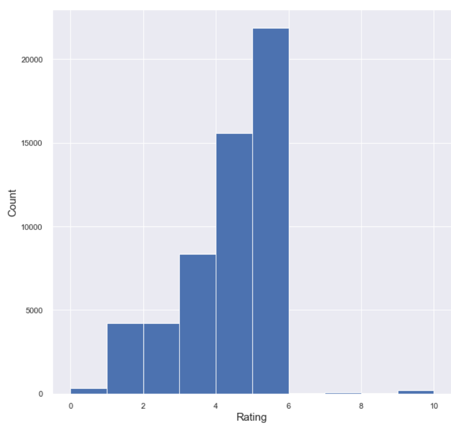


Figure 3: Rating Distribution on Re-scaled Data

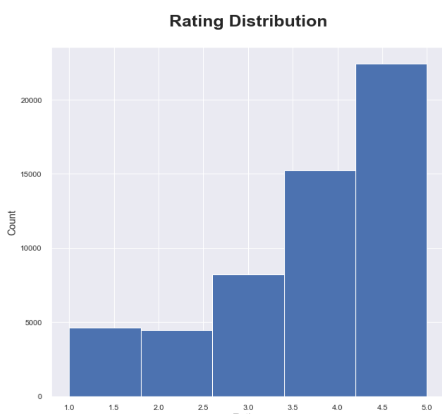


Figure 4: Hotel Locations (Lat, Long)

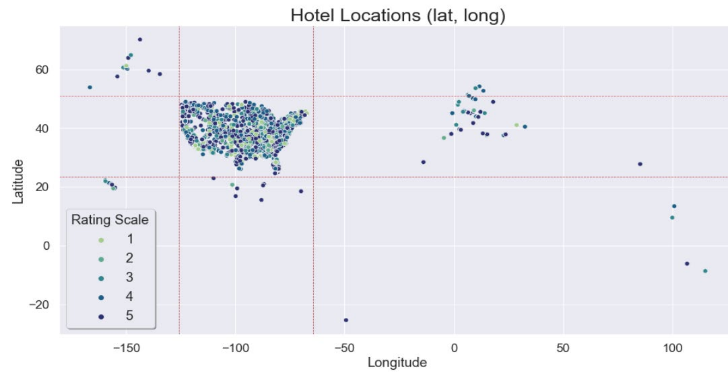


Figure 5: U.S. Hotel Locations (Lat, Long)

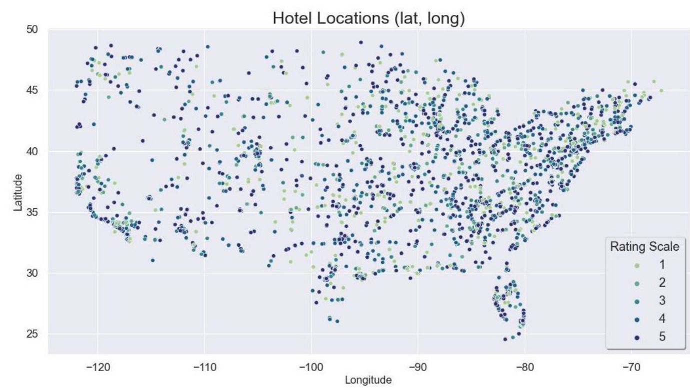
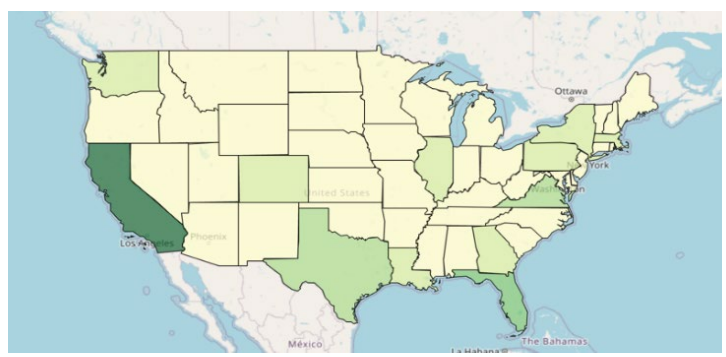


Figure 6: Heatmap (Lat, Long)



Appendix 2: EDA Exhibits

Figure 7: Average Rating by Hotel

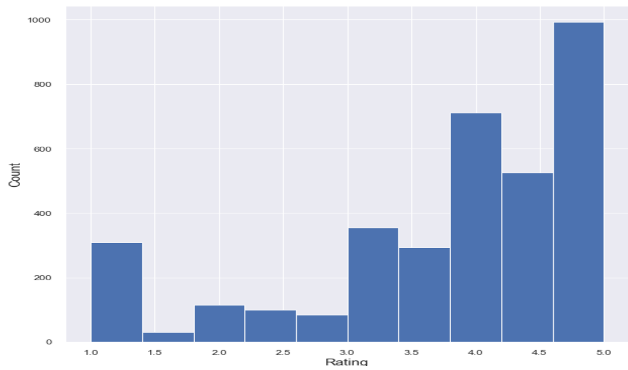


Figure 8: Average Rating by Hotel, for more than 5 reviews

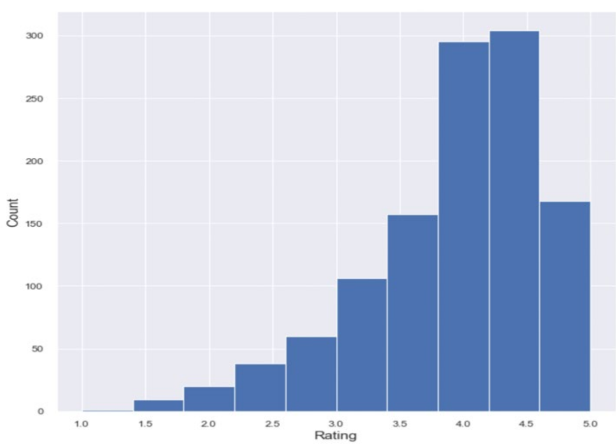


Figure 9: Hotel Review Category (by Rating)

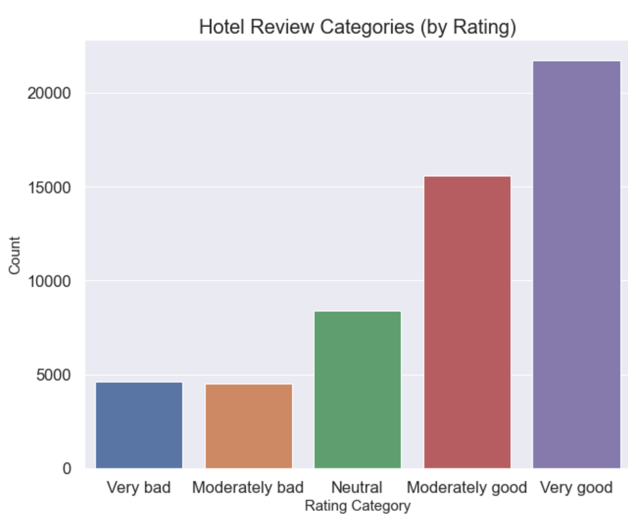


Figure 10: Hotel review Categories (Three Categories)

Figure 11: Hotel Reviews by Month

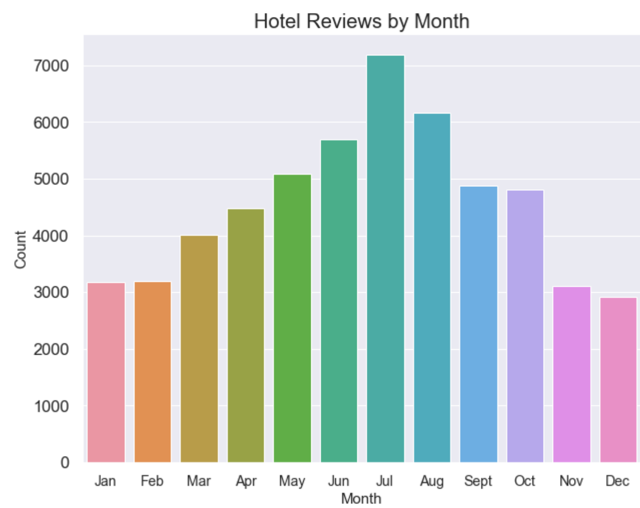


Figure 12: Ratings Categories by Month

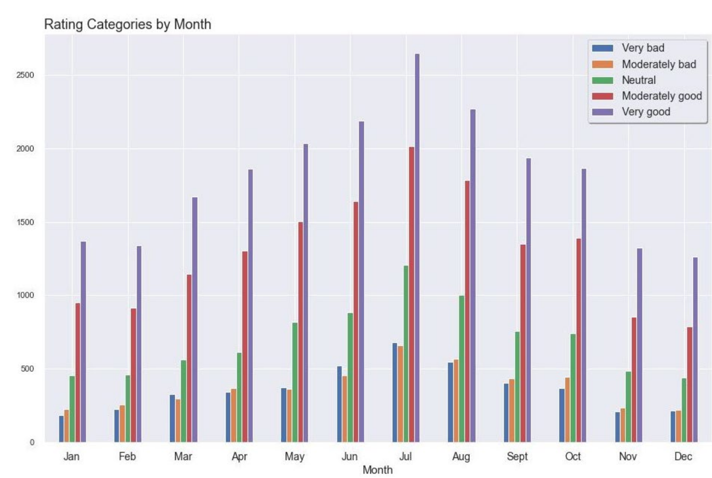


Figure 13: Review Word Clouds

Figure 14a: Very-good Reviews vs. Very-bad Reviews

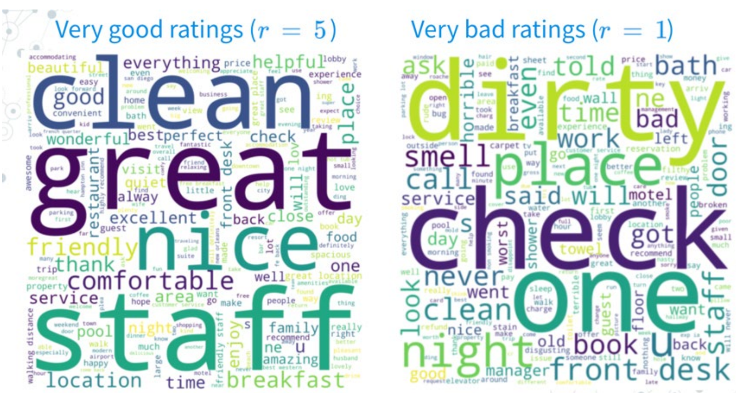


Figure 15b: Five-Class Word Clouds

Figure 16: Reviews Word Clouds for top hotels & bottom hotels

Figure 17: Review Summarization

```
get_summarization('Hampton Inn & Suites Warren')
```

the staff is extremely friendly and there's free breakfast in the morning. wonderful beds, very helpful staff and great breakfast.great the staff welcomed me as the guest of the day! the is nice, seems fairly new, or recently updated, and as we've found at most other hilton brand the staff is wonderful.

```
get_summarization('The Inn On Negley')
```

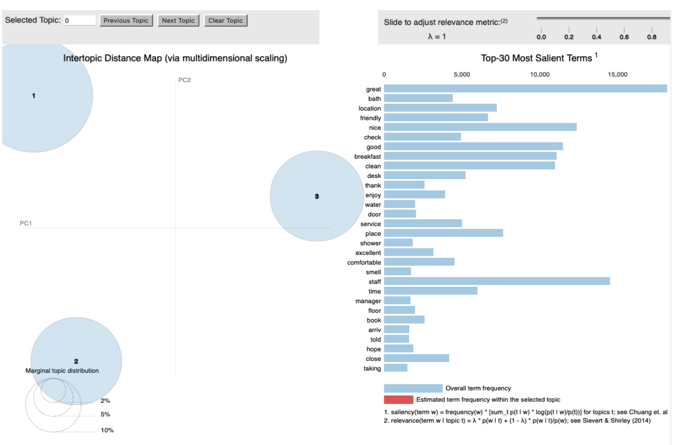
hot breakfast was excellent and hosts very friendly.great inn close to downtown french toast is amazing.a wonderful and relaxing only ed one night, but checked in at 1:00 pm and left at 11:00 the next day so it wasn't just in to sleep and out again.

```
get_summarization('Fiesta Inn and Suites')
```

we struggled to get them to even give us clean towels from the that they never cleaned, as they said on the 3rd floor they clean it only time week, bad service, at first they told us that we only had reservation for one and it was not beds as we needed it, the smelled exaggeratedly of tobacco, only the location of the was good, but overall disaster, will not come back !!!

Figure 18: LDA Topics Visualization with Word cloud

Figure 19: LDA Topic segmentation



Appendix 3: Modelling and Model Outcomes

Table 1: Regression Model Performance

Model	R-squared	Mean squared error	Explained variance score
Linear Regression	-0.482189	2.438575	-0.482032
Decision Tree Regressor	0.285970	1.174760	0.286032
Random Forest Regressor	0.308239	1.138122	0.308301
Nearest Neighbors Regressor	-0.790280	2.945463	0.165693
Multi-layer Perceptron	0.312164	1.131665	0.312384
Support Vector Machine	0.571882	0.704362	0.575777

Table 2: Model Accuracy - 5-Class Classifiers

Model	**Accuracy**
Decision Tree	0.40
Random Forest	0.52
Bagging Classifier	0.45
Logistic Regression	0.51

Table 3: Model Accuracy - 3-Class Classifiers

Model	**Accuracy**
Decision Tree	0.68
Random Forest	0.75
Bagging Classifier	0.74
Logistic Regression	0.78

```
5-Class Classification Performance
------------------------------------
Decision Tree Accuracy:         0.40
Random Forest Accuracy:         0.52
Bagging Classifier Accuracy:    0.45
Logistic Regression Accuracy:   0.51
```

```
3-class Classification Performance
------------------------------------
Decision Tree Accuracy:         0.68
Random Forest Accuracy:         0.75
Bagging Classifier Accuracy:    0.74
Logistic Regression Accuracy:   0.78
```

Table 4: Binary Classifier Performance (Review Text Only)

Metric	**Decision Tree**	**Random Forest**	**Bagging Classifier**	**Logistic Regression**
Precision	0.83	0.83	0.83	0.80
Recall/TPR	0.83	0.88	0.86	0.93
F1 Score	0.83	0.85	0.84	0.86
Accuracy	0.77	0.80	0.79	0.80
AUROC	0.76	0.83	0.81	0.84

Table 5: Binary Classifier Performance (Title Text Only)

Metric	**Decision Tree**	**Random Forest**	**Bagging Classifier**	**Logistic Regression**
Precision	0.83	0.83	0.83	0.80
Recall/TPR	0.83	0.88	0.86	0.93
F1 Score	0.83	0.85	0.84	0.86
Accuracy	0.77	0.80	0.79	0.80
AUROC	0.76	0.83	0.81	0.84

Figure 20: ROC Curve (Review text only)

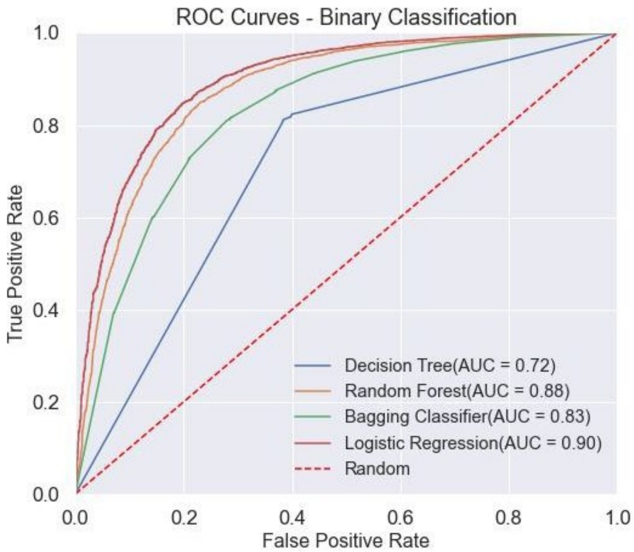


Figure 21: ROC Curve (Title text only)

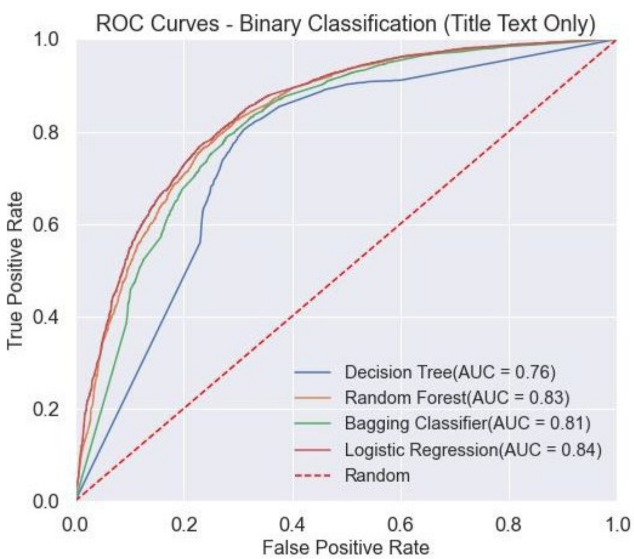


Figure 22: Precision-Recall Plot – Binary Classifiers (Review Text Only)

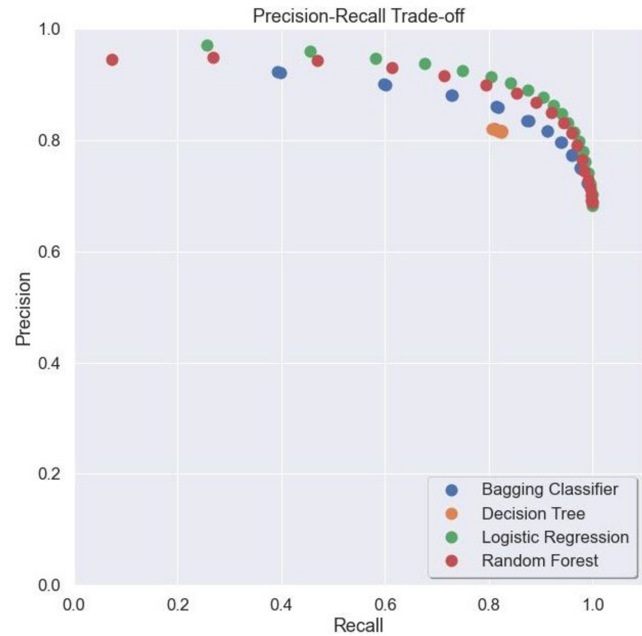


Figure 23: Precision-Recall Plot (Title text only)

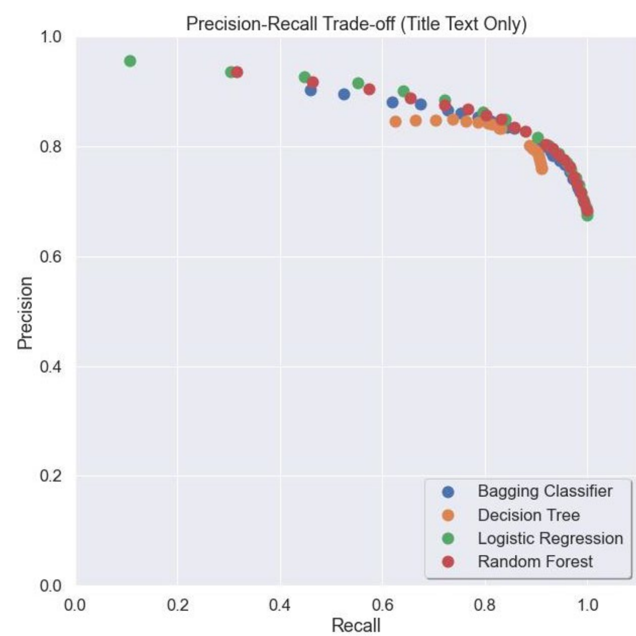


Figure 24: Review Compound Polarity vs. Title Compound Polarity (by individual review)

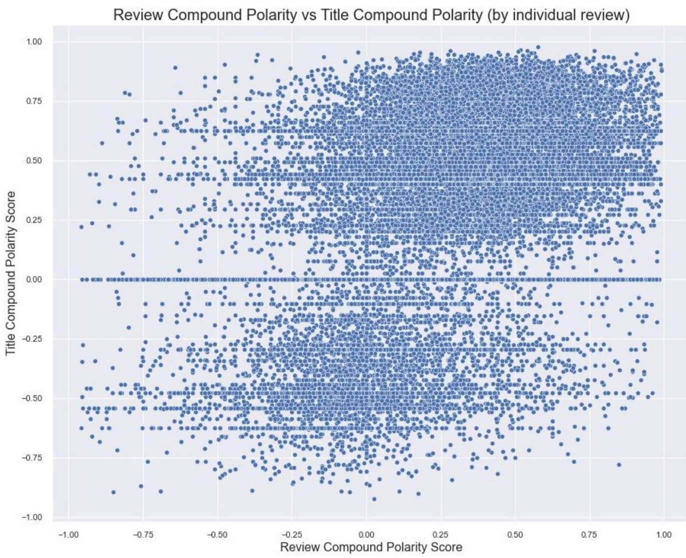


Figure 25: Review Compound Polarity vs. Title Compound Polarity (grouped by hotel)

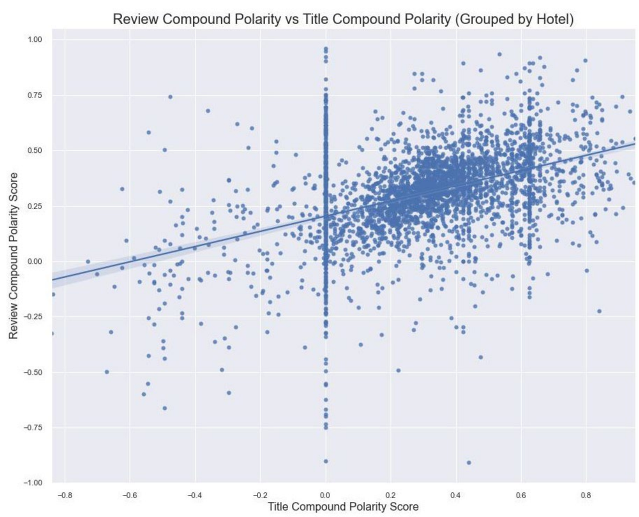


Figure 26: Review Compound Polarity vs. Average Rating (grouped by hotel)

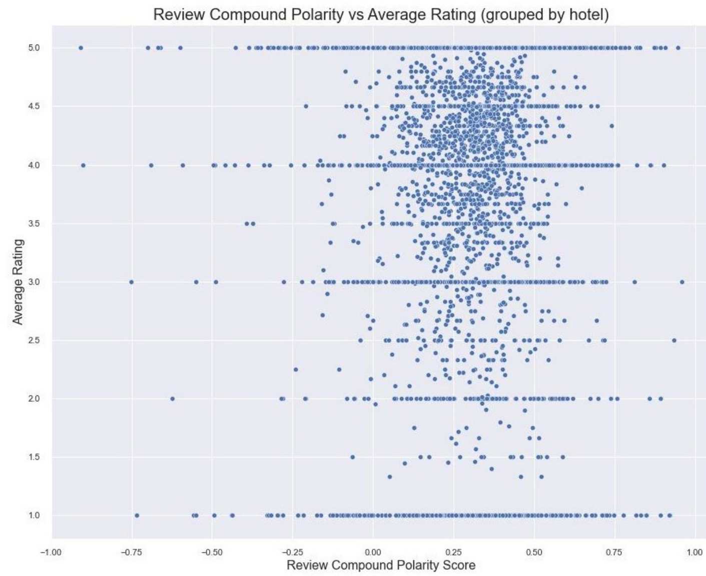


Figure 27: Title Compound Polarity vs. Average Rating (grouped by hotel)

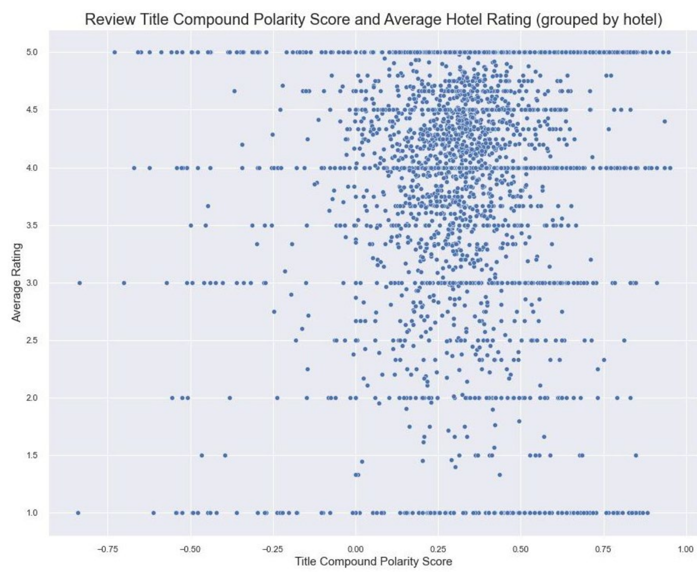


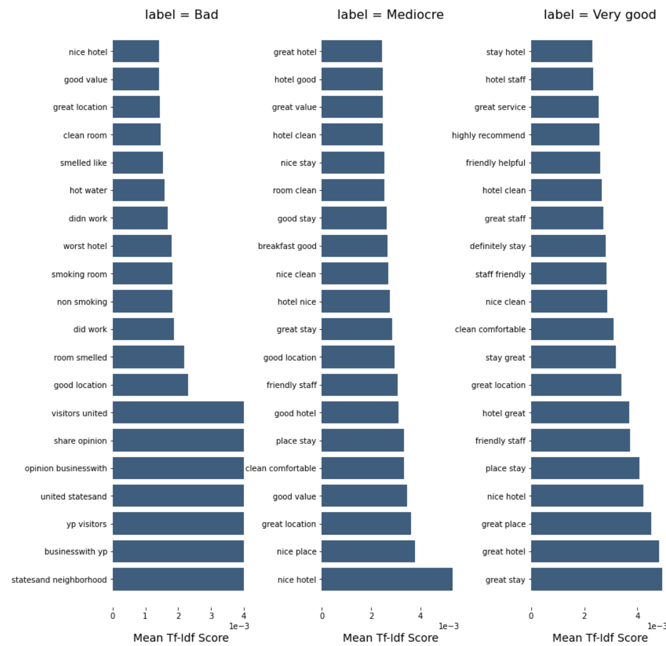
Table 6: Precision-Recall Pairs for various thresholds (review text only)

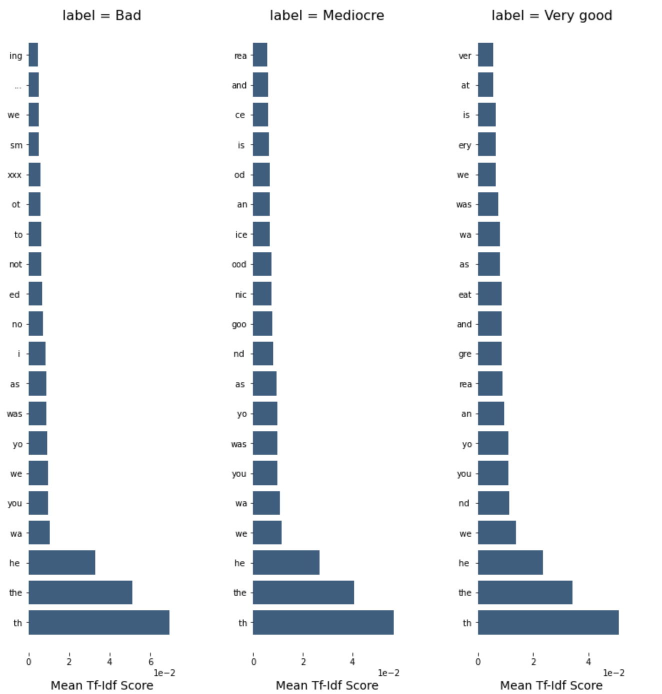
	model_label	threshold	precision	recall
60	Logistic Regression	0.0	0.675296	1.000000
61	Logistic Regression	0.05	0.679974	0.999527
20	Random Forest	0.0	0.683733	0.998580
62	Logistic Regression	0.1	0.689129	0.997003
21	Random Forest	0.05	0.698813	0.993374
63	Logistic Regression	0.15000000000000002	0.704754	0.991481
64	Logistic Regression	0.2	0.717640	0.988326
22	Random Forest	0.1	0.715134	0.987695
40	Bagging Classifier	0.0	0.717207	0.985013
65	Logistic Regression	0.25	0.730287	0.983278
41	Bagging Classifier	0.05	0.724026	0.981701
23	Random Forest	0.15000000000000002	0.732242	0.980596
66	Logistic Regression	0.30000000000000004	0.742631	0.977757
24	Random Forest	0.2	0.745501	0.973655
42	Bagging Classifier	0.1	0.741959	0.971604
67	Logistic Regression	0.35000000000000003	0.757190	0.967661
43	Bagging Classifier	0.15000000000000002	0.754898	0.966398
25	Random Forest	0.25	0.764272	0.965136
68	Logistic Regression	0.4	0.770683	0.959615
44	Bagging Classifier	0.2	0.766941	0.955198

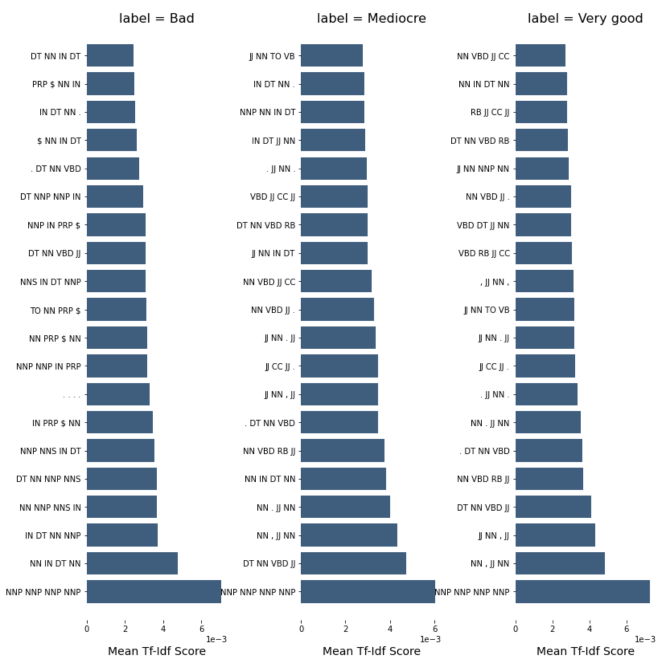
Table 7: Precision-Recall Pairs for various thresholds (title text only)

	model_label	threshold	precision	recall
60	Logistic Regression	0.0	0.683346	1.000000
20	Random Forest	0.0	0.688158	0.999844
21	Random Forest	0.05	0.689061	0.999219
61	Logistic Regression	0.05	0.703264	0.999063
22	Random Forest	0.1	0.692266	0.997814
23	Random Forest	0.15000000000000002	0.700186	0.996877
62	Logistic Regression	0.1	0.721801	0.996097
24	Random Forest	0.2	0.712864	0.995004
63	Logistic Regression	0.15000000000000002	0.741563	0.991413
25	Random Forest	0.25	0.729016	0.991257
40	Bagging Classifier	0.0	0.722861	0.990788
41	Bagging Classifier	0.05	0.723705	0.990476
64	Logistic Regression	0.2	0.762152	0.986573
26	Random Forest	0.30000000000000004	0.745919	0.984543
65	Logistic Regression	0.25	0.779197	0.981265
27	Random Forest	0.35000000000000003	0.765301	0.980016
42	Bagging Classifier	0.1	0.749342	0.977361
43	Bagging Classifier	0.15000000000000002	0.750750	0.976737
66	Logistic Regression	0.30000000000000004	0.797672	0.973770
28	Random Forest	0.4	0.790994	0.970804

Appendix 4: Feature Selection Exhibits




label = Bad
ing
...
we
sm
xxx
ot
to
not
ed
no
i
as
was
yo
we
you
wa
he
the
th
Mean Tf-Idf Score
1e−2
label = Mediocre
rea
and
ce
is
od
an
ice
ood
nic
goo
nd
as
yo
was
you
wa
we
he
the
th
Mean Tf-Idf Score
1e−2
label = Very good
ver
at
is
ery
we
was
wa
as
eat
and
gre
rea
an
yo
you
nd
we
he
the
th
Mean Tf-Idf Score
1e−2


label = Bad
DT NN IN DT
PRP $ NN IN
IN DT NN .
$ NN IN DT
. DT NN VBD
DT NNP NNP IN
NNP IN PRP $
DT NN VBD JJ
NNS IN DT NNP
TO NN PRP $
NN PRP $ NN
NNP NNP IN PRP
. . . .
IN PRP $ NN
NNP NNS IN DT
DT NN NNP NNS
NN NNP NNS IN
IN DT NN NNP
NN IN DT NN
NNP NNP NNP NNP
Mean Tf-Idf Score
1e−3
label = Mediocre
JJ NN TO VB
IN DT NN .
NNP NN IN DT
IN DT JJ NN
. JJ NN .
VBD JJ CC JJ
DT NN VBD RB
JJ NN IN DT
NN VBD JJ CC
NN VBD JJ .
JJ NN . JJ
JJ CC JJ .
JJ NN , JJ
. DT NN VBD
NN VBD RB JJ
NN IN DT NN
NN . JJ NN
NN , JJ NN
DT NN VBD JJ
NNP NNP NNP NNP
Mean Tf-Idf Score
1e−3
label = Very good
NN VBD JJ CC
NN IN DT NN
RB JJ CC JJ
DT NN VBD RB
JJ NN NNP NN
NN VBD JJ .
VBD DT JJ NN
VBD RB JJ CC
, JJ NN ,
JJ NN TO VB
JJ NN . JJ
JJ CC JJ .
. JJ NN .
NN . JJ NN
. DT NN VBD
NN VBD RB JJ
DT NN VBD JJ
JJ NN , JJ
NN , JJ NN
NNP NNP NNP NNP
Mean Tf-Idf Score
1e−3