Sentiment Analysis on Trip Advisor Hotel Review

Walid El Kassem

Introduction: Customer’s satisfaction is very important for the service industry. For this reason, it is necessary to determine the emotional state of the customer's thoughts. In this work I will create several machine learning models trained on trip advisor hotel reviews that will predict the sentiment of the customer based on the review

1 Collecting Data

The dataset was collected from Kaggle. It consists of 20491 reviews that are labeled on a scale from 1 to 5

	Review	Rating
0	nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took adv...	4
1	ok nothing special charge diamond member hilton decided chain shot 20th anniversary seattle, sta...	2
2	nice rooms not 4* experience hotel monaco seattle good hotel n't 4* level.positives large bathro...	3
3	unique, great stay, wonderful time hotel monaco, location excellent short stroll main downtown s...	5
4	great stay great stay, went seahawk game awesome, downfall view building did n't complain, room ...	5
...	...	...
20486	best kept secret 3rd time staying charm, not 5-star ca n't beat, time stayed increased esteem, b...	5
20487	great location price view hotel great quick place sights.directly street space needle downtown t...	4
20488	ok just looks nice modern outside, desk staff n't particularly friendly, corridors dark smelt st...	2
20489	hotel theft ruined vacation hotel opened sept 17 2007 guests week, happy stumble scouting hotels...	1
20490	people talking, ca n't believe excellent ratings hotel, just n't, yes patricia extremely helpful...	2

20491 rows × 2 columns

Figure 1: Screenshot of the Dataframe

2 Data Analysis & Creating new labels

After analysis the data, we figured out that the data doesn’t contain any missing entries and has no duplicates. We assigned a new labels for the reviews: positive, negative & neutral such that the positive reviews are those with rating > 3 and the neutral are those with rating equal 3 and the negative are those with rating <3. The new labels made the dataset unbalanced

Rating	Counts
positive	15093
negative	3214
neutral	2184

Table 1: The frequency of the created labels

That forced us to cut off 11879 postive reviews to balance the data in order to avoid overfitting the model.

Rating	Counts
positive	3214
negative	3214
neutral	2184

Table 2: The Frequency of the created labels after balancing the dataset

3 Cleaning the data

Using the NLTK , RE (regular expression) & Spacy libraries we cleaned the data by doing these 10 Steps:

1. lower case
2. remove html and urls
3. remove emojis
4. remove non-ascii charachters
5. remove emails
6. remove punctuation
7. remove stopwords
8. remove numbers
9. Lemmatizing
10. stemming

4 Train & test the models

First we split the data into train (80%) and test (20%) sets. Then we created the word embeddings of the Reviews using the TF-IDF , Word2vec from Spacy & and also the universal-sentence-encoder-Transformer from the Tensorflow. For the TF-IDF approach we created seven Models provided by scikit-learn [GaussianNB, DecisionTreeClassifier, RandomForestClassifier, SVC, LogisticRegression, KNeighborsClassifier & BernoulliNB] and trained them on the TF-IDF word embeddings. We used the cross validation technique to get accurate results. The results are shown in the Table 3.

Model	Accuracy
Guassian Naive Bayes	60.2699 %
Decision Tree Classifier	55.0441 %

Random Forest Classifier	68.9358 %
Support Vector Classifier	74.7858 %
KNeighbors Classifier	57.4539 %
Logistic Regression	74.7568 %
Bernoulli Naive Bayes	67.7748 %

Table 3: Training Results of Scikit-Learn Models

After that we tried to optimize the KNeighborsClassifier by tuning the parameters. The accuracy of the Classifier jumped from 59% to 68%.

N_neighbors	Weights	Accuracy
5	distance	59.1991 %
8	distance	61.3468 %
12	uniform	64.3068 %
12	distance	64.0163 %
16	uniform	66.1637 %
16	distance	65.8735 %
20	distance	65.8154 %
24	distance	66.1056 %
30	uniform	66.8601 %
34	uniform	67.6146 %

Table:4 K-NeighborsClassifier with different parameters

Then we tried another approach by creating new word-embeddings using Word2vec from Spacy and retrained the 7 models. However, the accuracy of our Models decreased. The results are prese Table 5.

Model	Accuracy
Guassian Naive Bayes	60.1006 %
Decision Tree Classifier	49.6130 %
Random Forest Classifier	64.8607 %
Support Vector Classifier	70.4721 %
Logistic Regression	72.1749 %
KNeighbors Classifier	57.8560 %
Bernoulli Naive Bayes	59.4814 %

Table 5: Results of the Models using Word2vec from spacy

After that we created a deep learning model using Keras (Sequential Model). First, we encoded our train and test texts using Universal-sentence-encoder-multilingual-large which is a Transformer that is designed by GOOGLE. After that we created 3 layers for the model where the first layer has 256 neurons and the second layer has 128 and the third layer (output layer) has exactly 3 neurons. The activation function is softmax. At the end, we trained the model and tested it. The result is shown in the following Table.

Model	Accuracy
Keras	66.80 %

Table 6: Accuracy of Keras Model

5 Trainig without cleaning the data:

Out of curiosity, we trained the same 7 Models from scikit-learn and got a better results compared with the results of the same Models trained on the cleaned data. The results are presented in Table 7.

Model	Accuracy	Improvement
GaussianNB	60.53 %	+ 0.5 %
Decision Tree Classifier	56.52 %	+ 1.5 %
Random Forest Classifier	71.27 %	+ 2.2 %
Support Vector Classifier	76.95 %	+2.3 %
Logistic Regression	76.75 %	+2.3 %
KNeighbors Classifier	67.61 %	+ 10 %
BernoulliNb	69.52 %	+ 2 %

Table 7: Table 3: Training Results of Scikit-Learn Models without cleaning

6 Conclusion:

After getting a satisfying result with the logistic regression model (77% accuracy), we can now analyse the sentiment of the trip advisors customers using machine learning models and help by improving the hotels services.