Sentiment Analysis of 515K Hotel Reviews

JINGRONG TIAN, LING YANG

[1]Georgetown University

<jt1204@georgetown.edu>, <ly227@georgetown.edu>

Abstract. All the travel agencies nowadays manage to satisfy the artificial intelligence-driven characterization in order to hold its position in the rapidly consolidating market. Thus a precise and scientific algorithm of recommendation, comparison, and matching is needed for every agency. This paper intends to provide a hotel recommendation system by using the dataset from 515K Hotel Reviews Data in Europe, scraped from Booking.com. The aim of this hotel recommendation system is to predict and classify hotels to different bins using a combination of machine learning algorithms and computational linguistic embeddings.

Keywords: Recommendation. XGBoost. Logistic Regression. Random Forest.

1 Introduction

Booking a hotel is always an essential and personalized step of planning a trip. It involves many factors to make a decision, such as country, distance and weather. Every traveler researches and customizes his/her's itinerary to find the perfect balance among price, adventure and luxury. We wish to study the behavior and common characteristics of previous customers who used Booking's online service to implement a hotel recommendation system. The system may not only save customers time on researching millions of options, but also give personalized and satisfying hotels.

In this study, we include many linguistic feature engineering techniques, and three machine learning models to construct our system. Specifically, three models are Random Forest, Naive Bayes, and XGBoost. The goal here is to use a simple algorithm to achieve a competitive result, and choose the most appropriate model from the comparisons.

2 Related Work

The research by Shreyas R. Labhsetwar[1], et al explores the use of Artificial Neural Networks (ANN) powered by Google's Word2Vec skip-gram algorithm for customer sentiment analysis and review classification. The proposed model achieves a high test accuracy of 0.9248, with an average F1-Score of 0.925. They used unsupervised sentiment clustering effectively classifies the reviews into four distinct categories and enables the Hotel Management to work out the major problems experienced by the customers. This paper has provided a profound work on how to use unsupervised learning and neural network to process large textual dataset. While, deep learning neural networks can take a extreme long time to train, and the black-boxes in the process is hard to interpret, we intend to design a supervised learning method that can balance between accuracy and efficiency.

3 Dataset

The dataset, available on Kaggle, contains 515,000 customer reviews and scoring of 1493 luxury hotels across Europe. Meanwhile, the geographical location of hotels are also provided for further analysis.

The main csv file contains 17 fields. The description of each field is as below:

- Hotel_Address: Address of hotel.
- Review_Date: Date when reviewer posted the corresponding review.
- Average_Score: Average Score of the hotel, calculated based on the latest comment in the last year.
- Hotel_Name: Name of Hotel
- Reviewer_Nationality: Nationality of Reviewer

- Negative_Review: Negative Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Negative'
- ReviewTotalNegativeWordCounts: Total number of words in the negative review.
- Positive_Review: Positive Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: 'No Positive'
- ReviewTotalPositiveWordCounts: Total number of words in the positive review.
- Reviewer_Score: Score the reviewer has given to the hotel, based on his/her experience
- TotalNumberofReviewsReviewerHasGiven: Number of Reviews the reviewers has given in the past.
- TotalNumberof_Reviews: Total number of valid reviews the hotel has.
- Tags: Tags reviewer gave the hotel.
- dayssincereview: Duration between the review date and scrape date.
- AdditionalNumberof_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: Longitude of the hotel

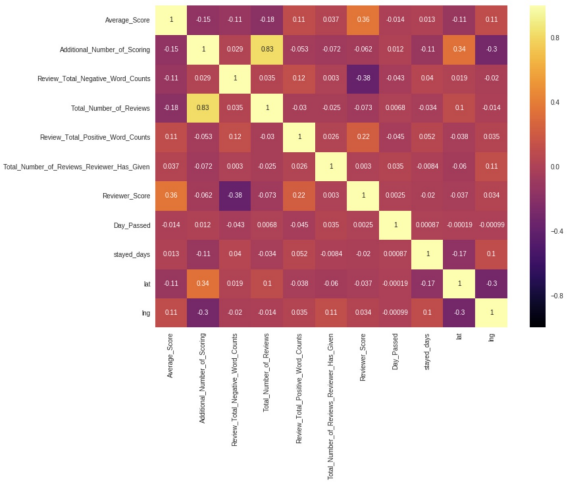
3.1 Data Transformation and Feature Engineering

1. To clean textual data and preprocess sentences, we first tokenize, lowercase, apply the Porter stemmer, and remove the punctuation tokens and stopwords.
2. New Feature Generation. We created additional column "Label"by dividing Reviewer_Scores into four classes, 3 with score>7.5, 2 with score > 5, 2 with score >2.5, 0 with score <2.5
3. Linguistic Feature Engineering Part We first added the TF-IDF (Term Frequency - Inverse Document Frequency) values for every word and every document. The TF-IDF metric solves word's relevance problem as: a. TF computes the classic number of times the word appears in the text. b. IDF computes the relative importance of this word which depends on how many texts the word can be found
4. Linguistic Feature Engineering Part 2. The next step consist in extracting vector representations for every review. The module Gensim creates a numerical vector representation of every word in the corpus by using the contexts in which they appear (Word2Vec).

   Each text can also be transformed into numerical vectors using the word vectors (Doc2Vec). Same texts will also have similar representations and that is why we can use those vectors as training features.

   We first have to train a Doc2Vec model by feeding in our text data. By applying this model on our reviews, we can get those representation vectors.

3.2 Data Correlation Analysis



It can be observed from the plot above that Reviewer_Score may have correlations with Average_Score ,Review_Total_Negative_Word_Counts ,Review_Total_Positive_Word_Counts.

4 Methodology

4.1 Random Forest

Random Forest is a supervised machine learning method that can be applied on both regression and classification problems. To generate predictions for clas-

2

sification problems, the algorithm builds multiple decision trees and takes the majority votes from decision trees. A random forest classifier is a meta-estimator that fits a number of decision tree classifiers on various subsamples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

The basic algorithm is as below:

Given a training set with n rows of data, in this project $n = 515,000$, which is then split-ted in X_train Y_train. For $b = 1, 2, ...B$ where $B =$ Number of sub-trees.

1. Assemble sets X_b, Y_b where $X_b \subseteq X_{train}, Y_b \subseteq Y_{train}$

2. Training regression trees on X_b, Y_b and collect the outputs

3. Take the average of all outputs.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x')$$

The value of B is adjustable. Normally for a large dataset, B is set to be around 1000 or higher.

4.2 XGBoost

XGBoost is short for Extreme Gradient Boosting, which refers to using ensembles of weak learners to make a prediction. XGBoost is based on this model. It is used for supervised learning problems, where we use the training data (with multiple features) x_i to predict a target variable y_i.

1. Creating regression trees and assemble the regression trees with their score

2. Create Objective for Tree Assemble. Assume there are k trees.

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i)$$

Objective function is

$$L = \sum_{i=1}^{n} \text{training loss}(y_i, \hat{y_i}) + \sum_{k=1}^{K} \text{Complexity of the Trees}$$

for *k=1 to K* **do**
  Propose $S_k = s_{k1}, ...s_{kl}$ by percentiles on feature k;
  Propose can be done per tree, or per split;
end
for *k=1 to K* **do**

$$G_{kv} = \sum_j g_j$$

  training loss ;

$$H_{kv} = \sum_j h_j$$

  Complexity loss;
end

Algorithm 1: XGBoost

4.3 Logistic Regression

Logistic regression is a linear classification method that learns the probability of a sample belonging to a certain class. Compared to XGBoost and Random Forest, logistic regression performs better when the number of noise variables is less than or equal to the number of explanatory variables and random forest has a higher true and false positive rate as the number of explanatory variables increases in a dataset.

1. The formula of the hypothesis

$$\sigma(Z) = \sigma(\vec{\beta}\vec{X}) \tag{1}$$

2. In order to map predicted values to scaled result, we use the activation function. The activation function maps any real value into another value between 0 and 1. Specifically for this project, we employed Sigmoid function.

$$\sigma(Z) = \frac{1}{1 + e^{-z}} \tag{2}$$

5 Results and Discussion

Three algorithms are implemented to build the recommendation system, including XGBoost, Logistic Regression, and Random Forest. Among the three algorithms, XGBoost gives the best result for both before and after the feature engineering work. Random Forest is most sensitive to NLP feature engineering, partly because numerically vectored textual data increased the quality and quantity of explanatory variables. Logistic regression has an insensitivity to the added NLP feature engineering work, as it is more a baseline algorithm for text related classification.

Accuracy Table Before NLP Feature Engineering

3

Model	Accuracy
Random Forest	0.715
XGBoost	0.754
Logistic Regression	0.738

Accuracy Table After NLP Feature Engineering

Model	Accuracy
Random Forest	0.746
XGBoost	0.767
Logistic Regression	0.736

6 Conclusion and Limitations

We have successfully implemented a hotel recommendation system using 515K Hotel Reviews in Europe dataset even though most of the data was anonymized which restricted the amount of feature engineering we could do. We ranked the problem at hand as a multi-class classification problem and maximized the probabilities with NLP feature engineering

The most important and challenging part of implementing the solutions was to create and extract meaningful features out of the 38 million data points provided to us. The exploration of data took a long time given the size of data and it helped us extract features that seemed to have high impact on predicting the hotel clusters.

Future works could focus on improving feature engineering by associating features together and creating pivot tables and pipelines for the later machine learning algorithms.

7 Bibliography

1. Labhsetwar, S. R. SENTIMENT ANALYSIS OF CUSTOMER SATISFACTION USING DEEP LEARNING. Research Journal of Computer Science (IRJCS), 6, 709-715