European Hotel Analysis

Aishwarya Gunasekar
University of North Carolina, Charlotte
Charlotte, NC
agunasek@uncc.edu

Alisha Gujarathi
University of North Carolina, Charlotte
Charlotte, NC
agujarat@uncc.edu

Dhiksha Ramkumar
University of North Carolina, Charlotte
Charlotte, NC
dramkuma@uncc.edu

Priyanka Sawant
University of North Carolina, Charlotte
Charlotte, NC
psawant2@uncc.edu

Shilpa Khandelwal
University of North Carolina, Charlotte
Charlotte, NC
skhande1@uncc.edu

Siddhant Gokule
University of North Carolina, Charlotte
Charlotte, NC
sgokule@uncc.edu

ABSTRACT

Hotel reviews are available in abundance today. This information can be utilized for analysis by both the hotel owners and the customers for various purposes. This project focuses on analyzing huge data set of hotel reviews and come up with visualizations that can communicate all the information clearly. The extracted information can be useful for the hotel owners to improve the ratings and reviews of their hotels by analyzing and prevailing their shortcomings. At the same time the customers choose the best hotel suiting their needs from the model trained on this data that predicts a best hotel given certain inputs. Organizations like the tourism department, travel bloggers, etc. can also utilize this data analysis for purposes suiting their needs.

CCS CONCEPTS

• **Big Data** → **Data Mining**; **Data Analysis**; • **Machine Learning** → *Regression*; *Classification*;

KEYWORDS

Data visualization, Visualization, Hotel reviews, Rating, Score

1 INTRODUCTION

Traveling is always an insight into a new side of the planet. Here we analyze the hotel data containing 515,000 customer reviews and scoring of 1493 luxury hotels across Europe. This is the exploratory data analysis for 515k Hotel reviews in Europe. By looking at the reviews of the Hotels we can get the information about the top rated hotels in Europe and can predict which hotel is best to stay while visiting Europe. Data-set used for this project is from Kaggle and can be found at Kaggle dataset[1]. This dataset contains 515,000 customer reviews and scoring of 1493 luxury hotels across Europe.

2 METHODOLOGY

First, the variables were studied and insights related to the dataset was gained by observing the distribution of the data and checking the range, frequency and shape of the data. During analysis, data preprocessing was done which involved activities like detecting and removing errors and inconsistencies, eliminating duplicates, convert numerical variable to categories, standardize the data and deriving new variables to be used to examine the target variables from different dimensions.

After the data was checked for tidiness, Business Use Cases were formed that would prove beneficial to the hotel user as well as the hotel owners. The hotel users can use these results to find out the best option available while they are visiting any country of their choice whereas the hotel owners or managers can request information that would allow them improve their ratings, so that more people visit their hotel.

The next step was to gain better understanding of the data by visualizing the data and any relation that may have existed between the data. This helped bolster the business use cases that were formulated and helped improve the solutions provided.

The business use cases formulated and visualizations done led to achieving multiple modeling techniques like regression, classification and sentiment analysis. These models provided solutions to the hypothesis that were formed in-order to help the users and hotel managers.

2.1 Data Preprocessing

The raw dataset contained number of inconsistencies like missing values, inconsistent date column format, NULL values and duplicate values which called for the data preprocessing. Data was cleaned following a systematic and planned approach before running any models on the data. For the purpose of data processing the CSV file was read in a R data frame. Preprocessing was performed on this data frame to remove the erroneous and null values, eliminating duplicates and correct date column format.

Order of data preprocessing:

(1) Find the total number of rows with missing values
(2)Remove Null values
(3) Eliminate duplicate values
(4) Uniformly format the date column to a single format

2.2 Hypotheses

Depending on the pre-processed data, there were few hypothesis that were formulated. These hypothesis were formed to improve the experience of both the hotel users and the hotel managers.

2.2.1 Correlation between Reviewer nationality and Reviewer score: To find the correlation between Reviewer nationality and

	Count Before Processing	Execution Command	Output	Count After Processing
Error and Inconsistencies Detection	515738	· *Load data in dataframe* Hotel_Reviews <- read.csv("C:/Users/shilp/Desktop/ Spring 2018/Big Data/Project/Hotel_Reviews.csv") · df <- Hotel_Reviews · nrow(df) · *#Number Of Rows With Missing Values* sum(!complete.cases(df))	3268	-
Null Records removal	515738	· *#Null Records Removal* df <-na.omit(df) · nrow(df)	512470	512470
Eliminating duplicates	512470	· *#Eliminating Duplicates* df <- distinct(df) · nrow(df)	511944	511944
Formatting Date Column	511944	· *#Formatting Date Column(m/d/y)* df$Review_Date <- format(df$Review_Date, format="%m/%d/%Y")	-	511944

Figure 1: Data Preprocessing Statistics

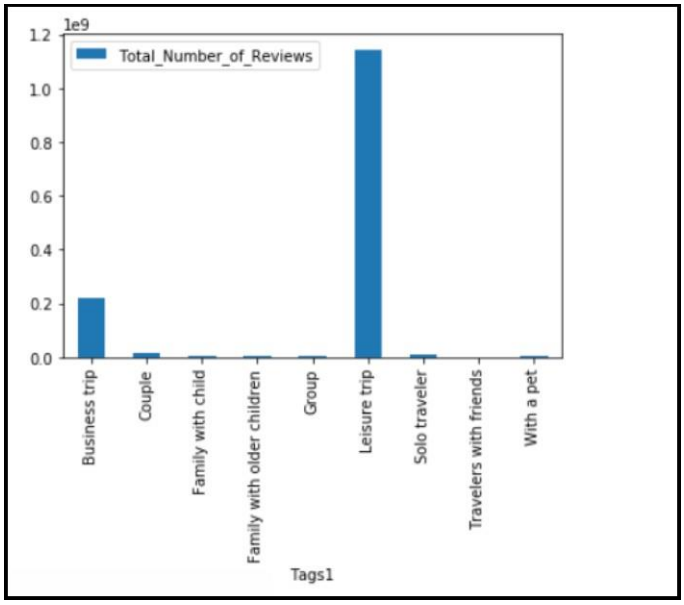

Figure 2: Relationship between Tags and Total number of reviews

Reviewer score, so the hotel managers can anticipate what the score of the reviewer could be and improve his hospitality towards the guest to get a better rating.

2.2.2 Predicting Average score of the Hotel based on Country, Hotel and Tags. Predict the average score of the Hotel based on the country of travel, hotel name and type of trip(classified by tags column) to help the User choose best hotel for his travel destination.

2.2.3 Suggest Best Hotel based on both Average score and Reviewer Score. Average score and reviewer score are very important features to suggest good hotels. Using these features in combined to suggest best hotels to the user.

2.2.4 Predict Recommendation. Predict the recommendation based on hotel profile. Using random forest model predicted weather a hotel is recommended or not.

2.2.5 Sentiment Analysis. A sentiment intensity analysis is performed over all the reviews to reflect the sentiments over people of a particular nationality, a specific hotel, a specific location. The model can also predict the sentiment of a particular hotel, nationality or location.

2.2.6 Finding important words/phrases that are indicative of reviewer score. Finding important words from the positive and negative reviews is important as they reflect the score that is given by the reviewer.

2.3 Data Visualization

2.3.1 Visualizing relationship between Tags and Total number of reviews. The above plot(refer figure 2) is tags and total no of reviews. So from the above plot we can observe that

(1) For Leisure trip we have the most number of reviews. So a person going for leisure trip will have more number of hotel and review options.
(2) Then, the business trip is the second most used tag and has more number of reviews for user to analyze the hotel.
(3) While other tags like travelers with friends have no reviews at all for any hotel.

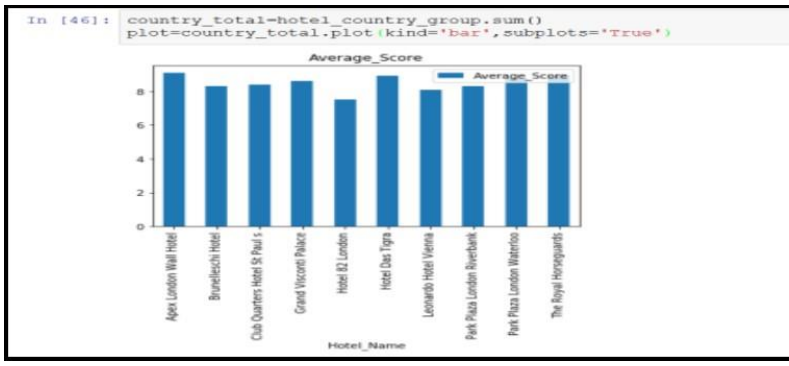

Figure 3: Average scores of hotels

2.3.2 Visualizing the average score of the hotels. Here (refer figure3) we are plotting the average rating of the hotels against their names. Following are the observations:

(1)Most of the hotels have average score 8 and above.
(2)Rare hotels have average score 7 or below

This is a primary observation of only 10 samples and it may be possible that for samples of entire dataset the observations may have slight variations.

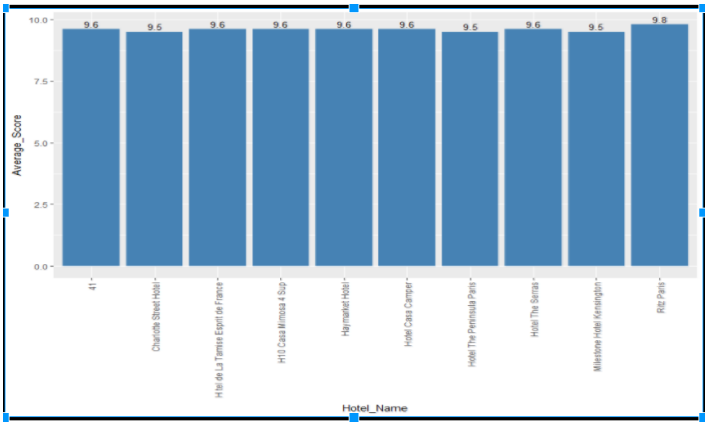

Figure 4: Top 10 rated hotel

2.3.3 Visualizing the top 10 rated hotels

. Following are the observations:(refer figure 4)

(1)Hotel Ritz is the one with highest Average Score of 9.8

(2)Among the top 10 highest rated hotels, least rated are Charlotte Street Hotel, Hotel The Peninsula Paris and Milestone Hotel Kensington with Average Score of 9.5

(3)None of the top 10 hotels have an Average Score of 9.5

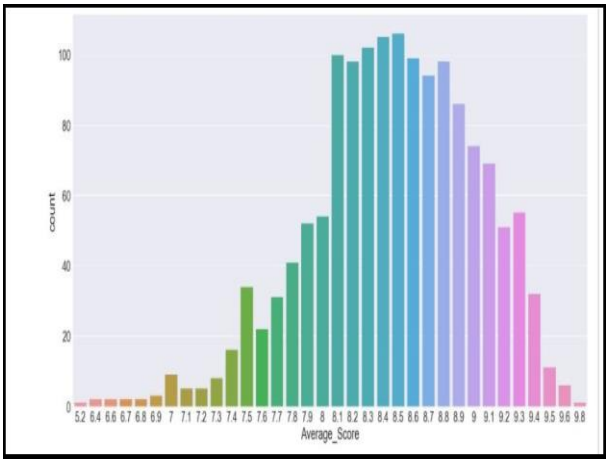

Figure 5: The range of Average score

2.3.4 Visualizing the range of Average score. From above figure (refer figure 5) we can observe that most of the Hotels average score lie in the range of 8.0 and 9.1 range

2.3.5 Visualizing Hotel Location with Average Score. Here the map (refer figure 6)shows the top hotels based on average score.

2.3.6 Correlation Matrix. In Figure 7, we see that there is a surprisingly high negative correlation between the Negative word counts and the Reviewer_score suggesting that a lot of people are tending to give positive reviews.

Figure 6: Hotel Location with Average Score

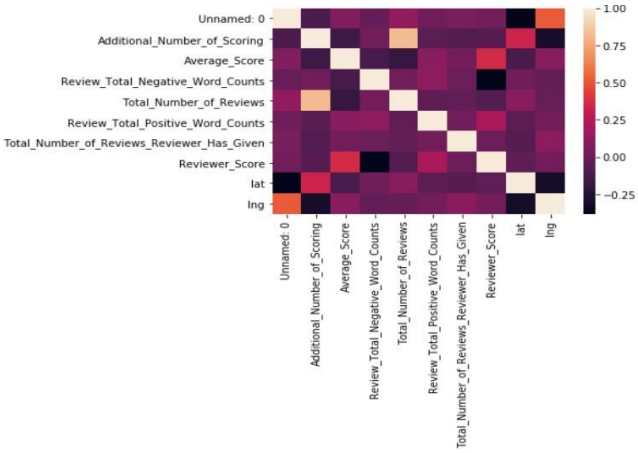

Figure 7: Correlation Matrix

2.4 Data Modeling

The hypothesis formulated in the previous section are provided with potential solutions using various modeling techniques.

2.4.1 ***Multiple Linear Regression.*** Regression was used to solve the hypothesis for finding the correlation between Reviewer nationality and Reviewer Score. This data model will help predict the average score a reviewer with a particular nationality would provide on visiting the hotel. Since the model of regression was run on XL-Miner[2], there were few constraints that the input data had to follow specified by the tool: a. There was a size constraint on the input dataset that could be fed to Multiple Linear Regression of 65,000 rows. This limited the input data set from 515k to only rows of 65,000. b. The reviewer nationality had 230 unique values and conversion of categorical variable to ordinal could only contain 30 maximum unique values. When these constraints had to be applied, there was a problem with random selection as there can be more of highly occurring countries which might make the prediction to be biased. In order to overcome this, to deal with the hypothesis, the

Australia	21502
United Arab Emirates	10170
Saudi Arabia	8903
Canada	7802
Israel	6527

Figure 8: Top 5 countries

top 5 most occurring countries were considered. This had fewer than 65,000 rows and also less than 30 unique variables. Performing this regression yielded that reviewers with Australian nationality gave the highest scores vs Guests from United Arab Emirates who gave the lowest among the 5 countries of Australia, Canada, Israel, Saudi Arabia and United Arab Emirates. This hypothesis was not only built using Multiple Linear Regression but also using two other models that were available in XLMiner. They were KNN Regression Model and Regression tree. All the three models gave almost the same results but Multiple Linear Regression was selected in order to facilitate the dependency between the reviewers score and nationality.

Limitations: The created model of multiple linear regression has a few limitations, such as the nationality. Though the dataset has the contents of 230 nationals, the only guests whose scores can be predicted are those of these 5 countries present in Figure 8. If a guest with a different nationality arrives, the score cannot be predicted.

Result: The MLR model had an average error of 6.01777294662043E-14 and RMSE error of 1.63, where as the RMSE error of the others were 1.64. This can be improved by improving the correlation between the decision and the target variable. The model stats are shown in figure 9.

Figure 10 shows the comparison of the regression models applied.

Result: The model has an average error of 6.01777294662043E- 14 and RMSE error of 1.65500675354277. This can be improved by improving the correlation between the decision and the target variable. The model stats are shown in figure 9.

2.4.2 ***Gradient Boosting Regressor.*** It is important to find out words/phrases that are suggestive of the reviewers score. Positive and negative words tend to affect the score. To regression model called Gradient Boosting Regressor is applied to the dataset to achieve this goal. The size of the data being quite large, it is divided into train and test sets. The regression model is trained on 20 percent of the data and validated on 80 percent of the data. The validation set is split into three parts for analysis. Applying a Gradient Boosting Regressor yields a model that tells us the importance of words. For instance, the presence of words like **negative**, **not** in the review reflect a lower reviewer score. On the other hand, words like **great**, **excellent** tend to reflect a higher reviewer score. The Gradient Boosting Regressor is slow to train but on the other hand it predicts very fast. GBR Unconstrained trees are susceptible to over fitting because they keep branching until they memorize the training data.

Regression Model

Input Variables	Coefficient	Std. Error	t-Statistic	P-Value	CI Lower	CI Upper	RSS Reduction
Intercept	8.871708	0.03211347	276.261218	0	8.808759	8.934657	696433.6
Reviewer_N	-0.20199	0.01055879	-19.1300762	3.7976E-80	-0.22269	-0.18129	1002.582

Residual DF	9998
R^2	0.035311
Adjusted R^2	0.035214
Std. Error Estima	1.655172
RSS	27390.47

Training Data Scoring - Summary Report

Total sum of squared errors	RMS Error	Average Error
27390.47	354277	6.02E-14

Figure 9: Regression Model Accuracy

Countries	MLR	kNN	Regression Tree
1=Australia	8.67	8.611568	8.611568
2=Canada	8.467	8.559719	8.559719
3=Israel	8.265	8.635452	8.63542
4=Saudi	8.063	7.836885	7.8537
5=UAE	7.86	7.873183	7.8731

Figure 10: Comparison

Results Figure 15 shows the list of words that affect the reviewer score with their weight.

2.4.3 ***Random Forest Model.*** Random forests are an ensemble learning method for classification and regression that operate by constructing a lot of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

The dependent or target variable is recommendation which explains whether the hotel is recommended or not based on hotel profile. Since the data is too huge for processing in R, we will consider data for one hotel. First, we selected the attribute from the dataset. We are not going to use all the attributes. Once we select the attributes, we can use Sample to split the data into 70percent train data and 30 percent test data. Now we ran Random Forest

```
Call:
 randomForest(formula = recommendation ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 10.31%
Confusion matrix:
     NO YES class.error
NO  148   9  0.05732484
YES  21 113  0.15671642
```

Figure 11: Random Forest

```
Confusion Matrix and Statistics

          Reference
Prediction  NO YES
       NO  156   3
       YES   1 131

               Accuracy : 0.9863
                 95% CI : (0.9652, 0.9962)
    No Information Rate : 0.5395
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9723
 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9936
            Specificity : 0.9776
         Pos Pred Value : 0.9811
         Neg Pred Value : 0.9924
             Prevalence : 0.5395
         Detection Rate : 0.5361
   Detection Prevalence : 0.5464
      Balanced Accuracy : 0.9856

       'Positive' Class : NO
```

Figure 12: RF-Train data

model. -> Figure 10 In this case, the number of variables tried at each split is based on the following formula. -1 is used as dataset contains dependent variable as well.

Formula: floor(sqrt(ncol(mydata) - 1))

Prediction and Confusion matrix for train data: Figure 11

Prediction and Confusion matrix for test data: Figure 12

Limitation:

1. Random Forests aren't good at generalizing cases with completely new data. A linear regression can easily figure this out, while a Random Forest has no way of finding the answer.

2. Random forests are biased towards the categorical variable having multiple levels (categories). It is because feature selection based on impurity reduction is biased towards preferring variables with more categories so variable selection (importance) is not accurate for this type of data.

Result: Accuracy of model for train data is 98 percent approx. which is a good accuracy. Accuracy of model for test data is 90 percent approx. which is a decent accuracy. Variable importance: top 5 important variable for prediction with random forest model. Most Important Variable: Figure 13

2.4.4 ***Sentiment Prediction.*** Preparing the training data for Sentiment Analysis: The Positive and Negative reviews for each row were first separated out and their intensities were computed in the range of -1.0 to +1.0. The resulting data consisted of each

```
> confusionMatrix(p2, test$recommendation)
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  58   9
       YES  2  45

               Accuracy : 0.9035
                 95% CI : (0.8339, 0.9508)
    No Information Rate : 0.5263
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8052
 Mcnemar's Test P-Value : 0.07044

            Sensitivity : 0.9667
            Specificity : 0.8333
         Pos Pred Value : 0.8657
         Neg Pred Value : 0.9574
             Prevalence : 0.5263
         Detection Rate : 0.5088
   Detection Prevalence : 0.5877
      Balanced Accuracy : 0.9000

       'Positive' Class : NO
```

Figure 13: RF-Test data

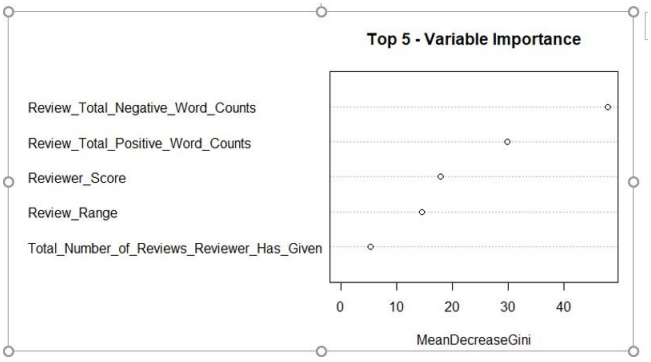

Figure 14: Most Important Variables

review associated with a numerical weight for it's positive review and a numerical weight it's negative review. There was however a small challenge that occurred, a review that had no negative comment was filled up with "No Negative" and a review that had no positive comment was filled up with "No Positive" in the original data. Since the intensity analyzer processed each individual word and then assigned it's appropriate weights, the "No Negative" and "No Positive" got whole weights of -1.0 and +1.0 which led to the model performing very badly as it tended to classify them as very negative or very positive reviews. To overcome this challenge we assigned a Neutral weight of 0.0 to all the "No Negative" and "No Positive" reviews. This improved the performance of the model. A new sentiment score was derived from the individual positive and negative weights to reflect the overall sentiment of the review. They were assigned into 6 categories in specific ranges: - -1.0 to

	Importance	Word
10650	0.045215	negative
10849	0.024920	not
10861	0.019890	nothing
7417	0.016455	great
6132	0.015441	excellent
12357	0.015180	positive
14367	0.013181	rude
15291	0.012839	small
10784	0.012818	no
1900	0.011925	bad
12278	0.011903	poor
5136	0.011882	dirty
9480	0.011608	location
11066	0.011512	old
11663	0.011145	perfect
16794	0.010882	tiny
14299	0.010828	room
1268	0.010658	amazing

Figure 15: Important Words

```
515143 {'neg': 0.0, 'neu': 0.674, 'pos': 0.326, 'compound': 0.6997}
515144 {'neg': 0.0, 'neu': 0.461, 'pos': 0.539, 'compound': 0.8934}
515145 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
515146 {'neg': 0.0, 'neu': 0.25, 'pos': 0.75, 'compound': 0.7184}
515147 {'neg': 0.379, 'neu': 0.0, 'pos': 0.621, 'compound': 0.34}
515148 {'neg': 0.0, 'neu': 0.349, 'pos': 0.651, 'compound': 0.8807}
515149 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
515150 {'neg': 0.379, 'neu': 0.0, 'pos': 0.621, 'compound': 0.34}
515151 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
515152 {'neg': 0.0, 'neu': 0.284, 'pos': 0.716, 'compound': 0.8769}
515153 {'neg': 0.0, 'neu': 0.483, 'pos': 0.517, 'compound': 0.6549}
515154 {'neg': 0.0, 'neu': 0.433, 'pos': 0.567, 'compound': 0.762}
515155 {'neg': 0.0, 'neu': 0.648, 'pos': 0.352, 'compound': 0.5859}
515156 {'neg': 0.0, 'neu': 0.476, 'pos': 0.524, 'compound': 0.5106}
515157 {'neg': 0.0, 'neu': 0.484, 'pos': 0.516, 'compound': 0.4927}
515158 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
515159 {'neg': 0.0, 'neu': 0.769, 'pos': 0.231, 'compound': 0.4019}
515160 {'neg': 0.0, 'neu': 0.56, 'pos': 0.44, 'compound': 0.6705}
```

Figure 16: The positive weights attached to each review

```
504311 {'neg': 0.056, 'neu': 0.859, 'pos': 0.085, 'compound': 0.431}
504312 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
504313 {'neg': 0.0, 'neu': 0.941, 'pos': 0.059, 'compound': 0.2023}
504314 {'neg': 0.094, 'neu': 0.906, 'pos': 0.0, 'compound': -0.5177}
504315 {'neg': 0.165, 'neu': 0.814, 'pos': 0.021, 'compound': -0.8634}
504316 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
504317 {'neg': 1.0, 'neu': 0.0, 'pos': 0.0, 'compound': -0.7096}
504318 {'neg': 0.078, 'neu': 0.879, 'pos': 0.043, 'compound': -0.5076}
504319 {'neg': 0.04, 'neu': 0.933, 'pos': 0.027, 'compound': -0.25}
504320 {'neg': 1.0, 'neu': 0.0, 'pos': 0.0, 'compound': -0.7096}
504321 {'neg': 0.206, 'neu': 0.688, 'pos': 0.106, 'compound': -0.4915}
504322 {'neg': 0.057, 'neu': 0.756, 'pos': 0.187, 'compound': 0.6868}
504323 {'neg': 0.158, 'neu': 0.842, 'pos': 0.0, 'compound': -0.4939}
504324 {'neg': 0.051, 'neu': 0.949, 'pos': 0.0, 'compound': -0.1027}
504325 {'neg': 0.297, 'neu': 0.65, 'pos': 0.053, 'compound': -0.8979}
504326 {'neg': 0.084, 'neu': 0.916, 'pos': 0.0, 'compound': -0.296}
504327 {'neg': 0.099, 'neu': 0.686, 'pos': 0.215, 'compound': 0.7752}
504328 {'neg': 0.185, 'neu': 0.815, 'pos': 0.0, 'compound': -0.7964}
```

Figure 17: The negative weights attached to each review

-0.7 - HIGHLY DISAPPOINTED, -0.7 to -0.4 - DISSAPOINTED, -0.4 to -0.1 - SAD, -0.1 to 0.2 - NEUTRAL, 0.2 to 0.5 - HAPPY, 0.5 to 0.8 - DELIGHTED, 0.8 and above - CAPTIVATED.

In Figure 17, there is an example of how these scores are computed for a few rows. The first column indicates the positive weight, the second column indicates the negative weight, the third column is the sentiment score computed using the two values and the fourth column is the resulting classified sentiment.

This data was then used to train the model using Support Vector Machine classification algorithm, which is used for predicting the sentiment if given a location, nationality or a particular average score or all of it. We plotted each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the classes very well. The classes are the various sentiments - Highly Disappointed, Disappointed, Sad, Neutral, Happy, Delighted and Captivated.

2.4.5 ***Neural Network Model***. Neural Network was used to predict the average score of a hotel when the hotel name, country in which hotel is located and the purpose of the trip is mentioned. Model was developed and trained in XLMiner with a maximum of 65,000 rows[4].

Top 30 hotels from the entire data set were chosen due to the large data size. Data was cleaned to extract the country of the hotel from it address as a part of preprocessing. Dummy variables of the

0.343	0.000	0.343	HAPPY
0.524	0.000	0.524	DELIGHTED
0.000	0.130	-0.130	SAD
0.165	0.104	0.061	NEUTRAL

Figure 18: Classiftcation ranges based on positive and negative weights

hotel name, hotel country and tags were created to eliminate the categorical variables and convert them into numeric values. Data was partitioned as Training Data (60%) and Validation Data(40%). XLMiner Neural Network's Manual Network was passed with training data for training the model.

The model was provided hotel name, hotel country and tags (which denotes purpose of the trip) as inputs and the average score as output. Trained model was then tested on the validation data to measure its performance. On the validation data it was observed that the model correctly predicts 1629 values out of total 1823 cases and has an overall error rate of **12.56%**. The confusion matrix image (Figure 19 gives a clear view about model performance on the validation data. Error report shown in Figure 20 provides detailed information about total cases belonging to a particular class and total errors in classifying cases of that class.

Limitations:

(1) Neural Networks need large datasets for training purposes
(2) Large dataset leads to increased training time for the model

Result:

(1) The model correctly predicted **2462** records out of total **2794** records with an error rate of **11.88%**
(2) For validation data the model only had **234** mis-classifications in total with an error rate of **12.56%**
(3) The hypothesis was modeled using Classification tree and got an error rate of **39.55%**(Refer Figure 21)

2.4.6 ***Classification treemodel.*** Decision tree learning is a method commonly used in data mining. Our goal is to create a

Confusion Matrix

	Predicted Class			
Actual Class	9.4	9.5	9.6	9.8
9.4	983	21	39	0
9.5	15	369	108	0
9.6	39	1	277	0
9.8	0	11	0	0

Figure 19: Confusion Matrix of Validation Data

Error Report

Class	# Cases	# Errors	% Error
9.4	1043	60	5.752637
9.5	492	123	25
9.6	317	40	12.6183
9.8	11	11	100
Overall	1863	234	12.56039

Figure 20: Error Report of Validation Data

Validation Data scoring - Summary Report (Using Best Pruned Tree)

Confusion Matrix

	Predicted Class			
Actual Clas	9.4	9.5	9.6	9.8
9.4	840	203	0	0
9.5	206	286	0	0
9.6	281	36	0	0
9.8	11	0	0	0

Error Report

Class	# Cases	# Errors	% Error
9.4	1043	203	19.46309
9.5	492	206	41.86992
9.6	317	317	100
9.8	11	11	100
Overall	1863	737	39.55985

Figure 21: Predicting average score using Classiftcation Tree

model that predicts the value of a target variable based on several input variables. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature and the child node gives the results of the decision[5]. The classification tree model here takes the features like average score and reviewer score and builds a classification tree that gives the hotel name as the output. The model will tell which hotel can have given average score and reviewer score based on the past data.

input: average score, reviewer score output: hotel name
procedure: Step1: Selected top 30 hotels based on average score

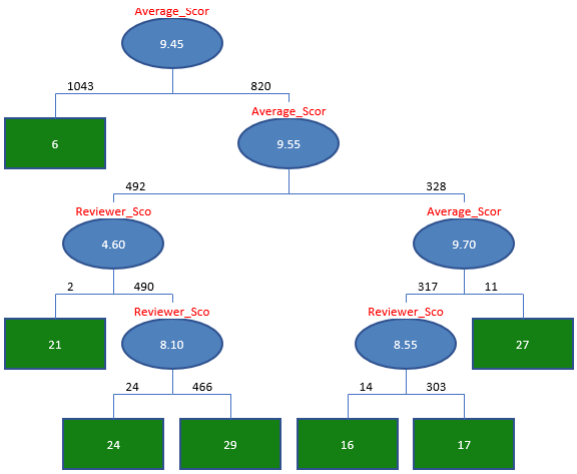

Figure 22: Classiftcation prune tree

Best-Pruned Tree Rules (Using Validation Data)

#Decision Nodes	6	#Terminal Nodes	7

NodeID	Level	ParentID	LeftChild	RightChild	SplitVar	plitValue/Se	Cases	sclassificati	Class	Node Type
0	0	N/A	1	2	erage_Sco	9.45	1863	0.90587	6	Decision
1	1	0	N/A	N/A	N/A	N/A	1043	0.466714	6	Terminal
2	1	0	3	4	erage_Sco	9.55	820	0.374732	17	Decision
3	2	2	5	6	viewer_Sc	4.6	492	0.222262	30	Decision
4	2	2	7	8	erage_Sco	9.7	328	0.118826	17	Decision
5	3	3	N/A	N/A	N/A	N/A	2	0.000716	21	Terminal
6	3	3	9	10	viewer_Sc	8.1	490	0.220472	30	Decision
7	3	4	11	12	viewer_Sc	8.55	317	0.112742	17	Decision
8	3	4	N/A	N/A	N/A	N/A	11	0	27	Terminal
9	4	6	N/A	N/A	N/A	N/A	24	0.00859	24	Terminal
10	4	6	N/A	N/A	N/A	N/A	466	0.209735	29	Terminal
11	4	7	N/A	N/A	N/A	N/A	14	0.003937	16	Terminal
12	4	7	N/A	N/A	N/A	N/A	303	0.107015	17	Terminal

Figure 23: Prune tree rules

and sampled the data. Step3: Converted hotel name columns from string to numeric values Step4: Partitioned data into train and test data. Step5: Build classification tree using XL miner on training data.

Given below in Figure 22 is a best prune classification tree:

Given below in Figure 23 are the rules obtained from the prune tree:

The major limitations of decision tree approaches include:

(1) Provide less information on the relationship between the predictors and the response.
(2)Biased toward predictors with more variance or levels.
(3)Can have issues with highly collinear predictors.
(4) Can have poor prediction accuracy for responses with low sample sizes.

3 CONCLUSION

The process of European Hotel Analysis was implemented by first pre-processing the data, which involved removing the occurrence of redundant values or NULL values to find insights related to the data to better understand the dataset. Later, hypothesis were formed to solve potential business use-cases which can help improve the business for the hotel managers and the experience of staying in a hotel for the guests. These hypothesis were solved by using data models formulated by machine learning techniques like Neural network and sentiment analysis, that could be created using XLMiner, Python and R. The data models created, throughout the projects can prove useful to the hotel business by providing suggestions to improve their hospitality and also to the tourists traveling to the area for a better experience for their stay.

A APPENDIX

A.1 Introduction

The 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' Review_Total_Negative_Word_Counts: 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' Review_Total_Positive_Word_Counts: Total number of words in the positive review. Reviewer_Score: Score the reviewer has given to the hotel, based on his/her experience

Total_Number_of_Reviews_Reviewer_Has_Given: Number of Reviews the reviewers 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: longitude of the hotel

A.2 Methodology

A.2.1 Data Preprocessing.

A.2.2 Business Use Cases. Reviewer Create reviewer profile Give a review/rating

Hotel Owners Create hotel profile on the review application View reviews and other information

Travelers Search listings for hotels Assess the reviews and make a decision USE CASE DESCRIPTIONS

Reviewer Use case number : 1 Use case name : Create reviewer profile Description : Reviewers who wish to review/rate the hotels they visited or stayed in in Europe will create a profile on the hotel review application.

Use case number : 2 Use case name : Give a review/rating Description : Reviewers who visited or stayed in the hotels in Europe can write a review on the hotel review application. The review can be inclined towards a positive or negative experience. They can also rate the hotel out of 10 by considering various factors that contribute to a pleasant stay

Hotel Owner Use case number : 3 Use case name : Create hotel profile on the review application

Description : The hotel owners who wish to get their hotels reviewed and want their hotels to be publicized to a larger traveler base will create a profile on the hotel review application.

Use Case Number: 4 Use Case Name: View reviews and other information

Description: Any hotel owner can continually track information on traveler feedback, scores, number of reviewers, number of positive/negative reviews, reviewer type, nationality and other such useful information to improve on their hotel business.

Traveler Use Case Number: 5 Use Case Name: Search listings for hotels

Description: Any traveler (Business/Leisure) intending to visit Europe will search among all the listed hotels on the hotel review application to make a well-informed decision for their stay.

Use Case Number: 6 Use Case Name: Assess the reviews and make a decision

Description: Any traveler (Business/Leisure), after going through a sufficient amount of hotel reviews, ratings, review dates, location, reviewer credibility scores and tags will make an overall assessment on the hotels and to come to a reasonable choice.

A.2.3 Data Modeling. Addition of New Column:

Exploring the data, it was observed that the average score given to a hotel was based only on the latest review received by a reviewer. This could affect the overall rating of the hotel by just one comment. Coming up with a solution for solving that issue(though vague), average of all the reviews were considered in creating a new feature of constructing a binary classification of if the hotel is good or bad.

The assumptions made to program this was that the columns with higher word count for positive is more positive than the negative if the word count if higher for positive word count. This assumption was based on the opinion made that when there were no negative comments present, the corresponding value was taken to be 0. So averaging on the hotels, we find if it has more negative or more positive reviews thus classifying it as good or bad. This would give the user an overall opinion of the hotel, if he does not want specifics of the reviews.

Addition of new columns - Positive weights, Negative weights, Sentiment score, Sentiment of the review under Sentiment Analysis.

ACKNOWLEDGMENTS

This project has been a success with support and contribution from all the individuals who had worked on this project. The authors would like to thank Dr. Li-Shiang Tsay for giving us the opportunity to work on this informative project. This was a great way to learn the concepts of Big Data and to apply Machine Learning techniques to large amount of data.

We would also like to extend our thank you to the teaching assistants who constantly helped solve any issues that we faced during the course of this project. Learning XL-miner for the first time, was difficult but it was made a smooth experience with the help of all our colleagues.

REFERENCES

[1] Dataset : https://www.kaggle.com/sampsonsimpson/exploring-515k-european-hotel-reviews

[2] Shmueli, G., Patel, N. R., & Bruce, P. C: Data mining for business intelligence: Concepts, techniques, and applications in Microsoft Office Excel with XLMiner. Hoboken, NJ: Wiley, 2010.

[3] A.H. Al-hamaami, S. H. Shahrour, " Development of an opinion blog mining system",Proceeding of the 4th International Conference on Advanced Computer Science Application and Technologies, pp. 74-79, 2015.

[4] Neural Networks and Statistical Models (1994) by by Warren S. Sarle

[5] The analysis of cases based on decision tree by Yurong Zhong.

[6] TomM. Micihen, " Machine Learning" in Carnegie Mellon University, January 2003

[7] Multiple Linear Regression: http://www.stat.yale.edu/Courses/1997-98/101/linmult.html

[8] Larose, Daniel T. Discovering Knowledge in Data: An Introduction to Data Mining. Wiley, 2014.