Group Members: Branden Diniz, Xuanyu Li, Dennis Silva, Chuxin Wei

**DS 501 Case Study 4: Data Science Shark Tank**

**Section 1 – Business Proposition**

No matter how versatile a product or service is, one cannot expect to sell it to everyone. But what if, in your business, one could maximize the likelihood that people will actually purchase your product or service? Every product or service has a target market, a group of people who are likely to purchase it. Understanding your target market and your targeted consumer is the key to provide them with services that fit their needs and desires. Expedia, Inc. (Expedia) has been longing to better understand their consumers on a deeper level, and provide them with the best service recommendations they can provide. Our team has answered their call.

Expedia is an online travel company. The company provides travel products and services to leisure and corporate travelers, offline retail travel agents, as well as travel service providers through numerous airlines, lodging properties, car rental companies, destination service providers, cruise lines, and other travel product and service companies. Within the business and personal services industry, Expedia is a leader in utilizing data driven technologies to provide their customers not only with the services they request but those that they do not even know they need until they use them. According to the very popular American business magazine Forbes, Expedia is ranked number fifty-seventh in the world for most innovative companies [1]. Ever since the company’s foundation in 1996, Expedia has collected petabytes of customer information through logs of customer behavior. As the company explains on the predictive modeling and analytics competition website Kaggle, Expedia wants to take the proverbial rabbit hole out of hotel search by providing personalized hotel recommendations to their users [2]. Expedia currently users only search parameters to adjust their hotel recommendations, but there aren’t enough customer specific data to personalize them for each user. Contextualizing customer data, our team has created an algorithm that can generate predictions based on the likelihood a customer will stay in one of one-hundred different hotel groups. With this algorithm, one can turn what may be a vacation disaster for some into a trip to paradise.

**Section 2 – Motivations**

When individuals, couples, or families are booking their dream vacation or weekend escapes, there can be hundreds, or even thousands, of hotels to choose from near every destination. A hotel is the hub for any vacation. One uses a hotel to relax, re-energize, and enjoy their company or solitude when they are not exploring their visited destinations. If one books a hotel that they believe fits their individual needs, yet one they arrive it is lackluster, it could spell disaster for them and their partners. If however the recommendation suffices or even exceeds their expectations, the quality of their vacation goes up and they are more likely to use Expedia for their future booking needs. This increases customer retention, overall company quality, and generated revenue.

In a world where one has access to the entire internet at the tips of their fingers, more and more individuals are looking to online booking websites for their travel needs. Aware of this trend, Expedia is a rapidly growing company in the online travel industry. Just last year Expedia purchased rival company Travelocity for $280 million. Not only that, but Expedia has also recently purchased an even larger online travel company Orbitz Worldwide for $1.33 billion [3]. As of May 2015, Expedia’s market capitalization, total market value of its shares, was $12.5 billion [1]. With these acquisitions and its new found growth, Expedia has acquired hundreds of thousands of new users. These users mean more data to better our data analysis techniques and more customers eager to spend their time and money vacations and travel. This is the time to understand Expedia’s target market and enhance customer loyalty by showing just how well Expedia can provide hotel recommendations. Taking a data science approach to the problem seems like the next logical step in this process. With so much data, finding the underlying structure in the data and removing noise is fundamental to providing accurate recommendations.

**Section 3 – Data Analysis**

Expedia has recently held a competition through the predictive modeling and analytics website Kaggle. Expedia challenges the site’s users to contextualize customer data and predict the likelihood a user will stay at a collection of one-hundred different hotel groups. The site contains four individual files: *sample\_submission.csv, destinations.csv, test.csv*, and *train.csv*. The files *test.csv* and *train.csv* contain the logs of customer behavior for millions of user event transactions. These user events are not equivalent to users, but each booking or clicking of a particular hotel cluster by a particular user. Hotel clusters are groups of hotels created by proprietary in-house algorithms created by the individuals at Expedia. These hotel clusters are based on characteristics such as historical price, customer star ratings, geographical locations relative to city center, and many more. These clusters are meant to serve as good identifies to which types of hotels people are looking to book, while avoiding outliers such as new hotels that don’t have historical data. As such, competitors are set to not predict which hotel users will book, but which of one-hundred hotel clusters they will book. The file *destinations.csv* contains the IDs of the destinations where the hotel search was performed by the user and a set of latent descriptions for each of one-hundred and fifty search regions. The file *sample\_submission.csv* is an example of a submission a competitor sends to Expedia for evaluation of their data analysis. This evaluation process will be described in later paragraphs.

Before creating any algorithms, our group wanted to first get a better understanding of the testing and training data themselves. Below is a list indicating the features within each of the customer behavior logs. Only 10,000 customer’s logs were sampled for this step.

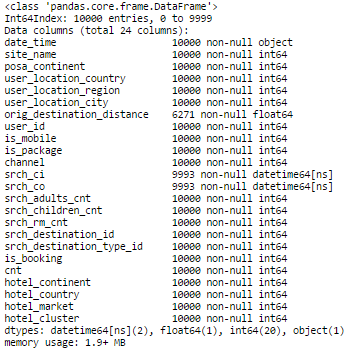
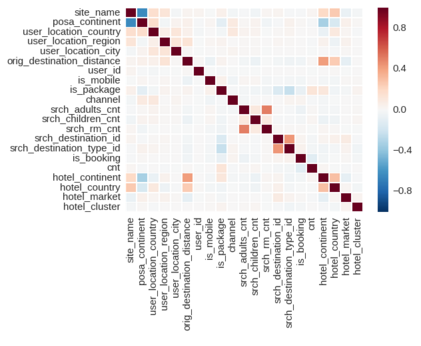


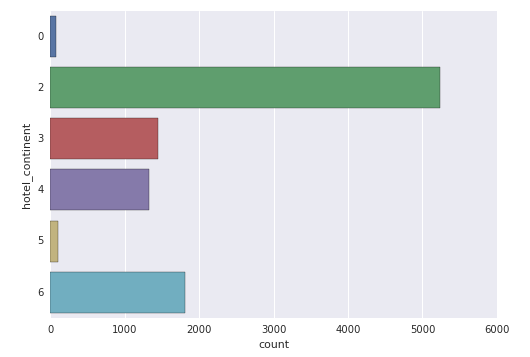
Table 1: The features of each customer behavior log stored in *training.csv* and *test.csv*. Note that *test.csv* neglects the feature ‘hotel\_cluster.’

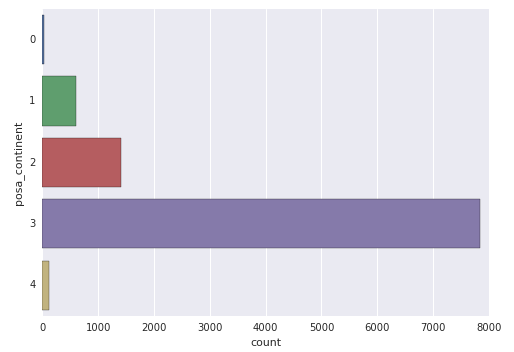
From this data we were able to view the types of features stored in the data. Expedia’s competition page in Kaggle also had short descriptions of each of these features. From this table we learned quite a bit. First we found that all features were numbers except for ‘data\_time,’ ‘srch\_ci,’ and ‘srch\_co.’ This meant if were wished to use some types of machine learning algorithms, we would need to preprocess the data to make these values usable. We also found that, although the data was already cleaned by Expedia, the data contained missing values. From the sample of 10,000 logs above one can see, for example, ‘orig\_destination\_distance’ only had 6271 entries. This meant our group would have to incorporate these missing values into our algorithm by filling them in or neglecting them all together. In addition, Expedia provides a limit to the amount of memory one can use for the competition. One can only use 8gb of memory at any given time for the algorithm, and reading 10,000 logs out of millions used approximately 1.9 MB. This mean our algorithm had to use memory wisely.

The Kaggle website allows users to share and distribute code of willing participants to all competitors. Other individuals began to investigate the features, and we incorporated some of their code that we found familiar to what we have done in previous case studies into our notebook to gain some more insight about the data. First we created a correlation matrix of the features.

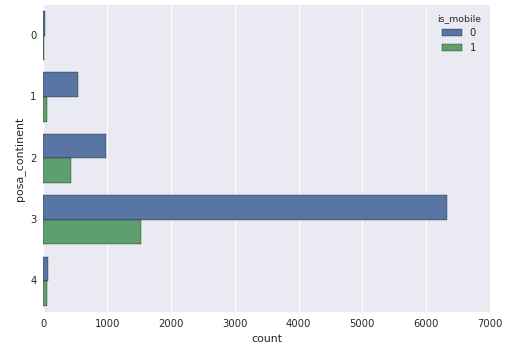
  
Figure 1: Correlations between different features within each customer behavior log.

It should first be noted that some correlations have very little meaning. For example ‘user\_id’ is meant to be an identifier of an individual and talking about its correlation between other variables is nonsensical. However we obtain some useful information. For example, ‘hotel\_continent’ (the continent the hotel is located on) has a positive correlation to ‘orig\_destination\_distance’ (physical distance between a hotel and customer at time of search) and a negative correlation to ‘posa\_continent’ (the continent where the booking was done). This means that individuals that tend to book on Expedia are going to hotels farther away from their home continent. To better visualize this type of information, we plotted some graphs.

  
Figure 2: Graph depicting counts of user events by the continents the hotel is located at.

  
Figure 3: Graph depicting counts of user events by the continents the hotel was booked from.

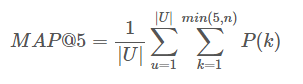
Although Expedia did not release the actual continent names for each ‘hotel\_continent’ or ‘posa\_continent’ value, one can see that most individuals are traveling to continent 2 for their travels, but most individuals are traveling from continent 3. We can break this information down even more, and figure out where individuals are using Expedia’s mobile booking or online booking applications by separating by the ‘is\_mobile’ feature.

  
Figure 4: Graph depicting counts of user events by continents the hotel was booked from and divided. The data is further divided by type of application used to book by the user.

The above graph shows that individuals from all continents tend to not utilize the mobile application of Expedia (indicated by a 0). Although our data is directly related to overall user hotel cluster recommendation, the information we found indicates there is an additional need to promote mobile booking on a global level.

With a better understanding of the features in the data we needed to pick a classifier. We decided to use a Random Forest Classifier. We decided on this type of classifier as Random Forests are known to be able to deal with samples (n) over a large number of features (p). Although there are a few million logs of customer data, with twenty-four distinct features we would need several hundred million logs or more to have a very accurate understanding of customer hotel booking behavior. The Random Forest Classification implementation for Python also allows us to generate and extract probabilities for which we can make predictions with.

Expedia requires competitors to provide up to five predictions of hotel clusters for each user event. These predictions reflect the likelihood that a user event will be held in a certain hotel cluster. The evaluator used by Expedia to judge how ‘good’ the algorithms do is called Mean Average Precision of 5 (MAP@5). The equation for MAP@5 is given by:



where |*U|* is the number of user events in the testing set, *P(k)* is the precision at cutoff k, and *n* is the number of predicted hotel clusters. This means that when an algorithm attempts to predict the most likely hotel clusters, one does not only get evaluated by if they predicted correctly in inclusion to the top five probabilities, but also the ranking or placement amongst those probabilities. A very clear and intuitive description of MAP@5 in the light of recommendations is described by a highly rated Kaggler in [4].

With all of this in mind, we moved through the hacking portion of our investigation. As a basis, we utilized an implementation of a low-memory version of a Random Forest Classifier that was shared by Kaggle user “Yair Beer” on the public Kaggle forums [5]. The code is similar to the algorithms our group generated for Case Study 3, but with the addition of utilizing a small amount of memory and converting our results into forms that can be read and evaluated by Expedia. After converting the skeleton of the code from a Python 3 environment to a Python 2 environment, we began machine learning and tuning the algorithms. The algorithm firsts reads in and converts the customer logs into a Pandas data frame and preprocesses the data by filling in missing feature values and removing NaNs. It then splits the *training.csv* file into yet another training/testing subset of 66% to 33% split respectively. We then conduct a Random Forest Classifier to produce probabilities of each user event stored in the training subset for each of the one-hundred hotel clusters. We then evaluate the predictions over the testing subset we created, and then create a submission file of the predictions generated for the user events contained in *testing.csv*. That file is then uploaded to Kaggles website for evaluation based on the left-out ‘hotel\_cluster’ information left out of *testing.csv*.

While tuning our algorithm we found that a Random Forest Classifier with 100 estimators and a max depth of 10 lead to our best performance. It is likely we would have increased our accuracy by increasing the number of estimators to 1000 or more and a max depth between 10-20, but the small computing power on our (older) laptops was a limiting factor.

**Section 4 – Results: Supporting our Proposition**

Based on the hidden hotel cluster data missing from the *test.csv* data file, our team titled ‘DSTeam’ had a MAP@5 of 0.12321. For comparison, the sample submission benchmark was 0.01744, the random guess benchmark was 0.02260, the most frequent benchmark was 0.05949, and the top scorer at the time of this writing was 0.49311. Because the hotel cluster data used to create these MAP@5 scores is not available to us, we used the testing subset data from our algorithm to generate the following graph.

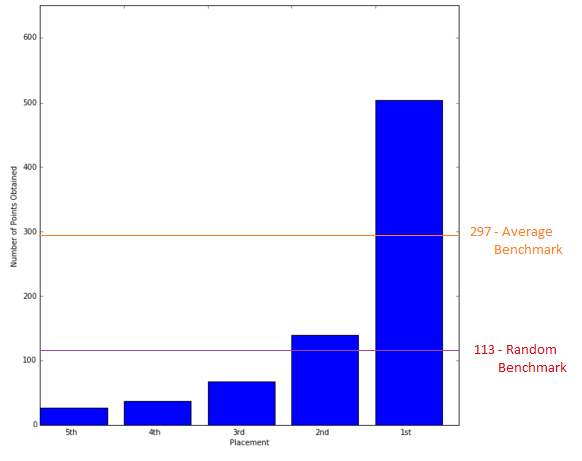


Figure 5: Points received from Mean Average Precision of 5 (MAP@5) over a random test sample of 5,000 user events. Lines represent average point totals for three different benchmarks given by Kaggle leaderboards.

As one can see, the graph indicates that, although we are currently not the best, our algorithm is above the most frequent benchmark for data generation. The graph would be more accurate if either our group could view the hidden cluster data or the current benchmark Expedia’s previous algorithms could produce.

**Citations:**

[1] Forbes. *Forbes Magazine*, 25 May 2016. Web. 25 Apr. 2016. <http://www.forbes.com/companies/expedia/>.

[2] "Expedia Hotel Recommendations." *Kaggle: The Home of Data Science*. N.p., 15 Apr. 2016. Web. 16 Apr. 2016. <https://www.kaggle.com/c/expedia-hotel-recommendations>.

[3] Tuttle, Brad. "What Expedia's Acquisition of Orbitz and Travelocity Means for Travelers."*Time. Time Magazine*, 12 Feb. 2015. Web. 25 Apr. 2016. <http://time.com/money/3707551/expedia-orbitz-impact-travelers/>.

[4] Schumacher, Aaron. "Facebook Recruiting Competition*." Alternate Explanation of Mean Average Precision -. Kaggle: The Home of Data Science*, n.d. Web. 25 Apr. 2016. <https://www.kaggle.com/c/FacebookRecruiting/forums/t/2002/alternate-explanation-of-mean-average-precision>.

[5] "Yair Beer." *Kaggle: The Home of Data Science*. N.p., n.d. Web. 18 Apr. 2016. <https://www.kaggle.com/mrbeer>.