# Expedia Hotel Recommendations Report

# Andrew Patterson

**Preliminaries**

* In the preliminary investigations, I wanted to understand at what size of the data and at what level the train set was – It was known it was event data but it was good practice to see what other interesting id levels there may be – for example a classifier could have easily been provided at the user level since all test user\_ids were in the train. Given the short amount of time I went ahead with an event level classifier
* Given the description, whereby the test set were booking only events, I discarded any non-booking events from the train data.
* I sampled the train data to 100,000 booked events to computational efficiency and speed going forward.

**Understanding the data**

* While most of the data was integer form, there were mostly encoding for categorical variables. From the I had two options – to binarize the data to use a regression based classifier or to use a different type of classifier and keep the data the same. Again, given the short time frame I thought I would use a Random Forest as the classifier and bypass any direct manipulation.
* For the date and datetime fields I pulled out key values (hours, day of week etc.) and discarded the dates themselves.
* For the latent variables in destinations.csv, I use PCA to cut down the 149 variables down to 6. This choice was made by looking at the cumulative variance the 6 new variables described the original data (~70%) which I thought was acceptable.

**Classification and testing**

* As said previously I decided to use a random forest as most of the variables were now encoded categorical variables. Rather than looking at a single y variable with 99 different segments, the better algorithm looked at for each event the probability of it falling into each segment. From there I could understand what were the top X segments for each event.
* I tested using both a probability cutoff and a ‘top x’ cutoff, and given the success criteria (MAPE@5) I found that giving the top 5 segments for every event.
* In the test/train split of the sampled test data the MAPE@5 score was 0.2074
* The score from the Kaggle website (screenshot attached) was 0.18794

**Possible Further investigations / improvements**

* In terms of the data handling, understanding the data at a user level would be the most beneficial course of action and would generate the best outcomes, which would be my next investigation
* Potentially splitting out the categorical variables to binary variables at each level, and using PCA to reduce this down would take out a lot of the noise in the data and may improve overall accuracy.
* For the classification, to make it more robust, I would implement a cross validation step to make sure the test/train sets do not vary and to understand the distribution of the MAPE@5 score.
* Furthermore, an implementation in something more resilient to more data (rather than local analysis using pandas) would mean that the whole train dataset would be able to be used