# Introduction

The goal of this project is to determine a reliable way to predict if a user will click on an element of a website, in this case a hotel listing from Trivago. What makes someone want to click on one image or link over another?

Companies use click information in a number of ways to increase their revenue. Including but not limited to marketing and advertising strategies, when to offer promotion and on which types of products. In addition, understanding user preferences (clicks) helps companies like Trivago to know what order to show their listings and be able to create personalized recommendations to each user.

This project frames the problem as wanting to know if a user will click or not on an item. In order to determine this, two different classification models are tested to determine which is best in addressing this problem.

This project is a work in progress as there are a number of additional recommendations and further work that can be done as outlined in the conclusion.

# Data Collection

The dataset used is a dataset provided by Trivago. The dataset is 2GB and contains 15,932,992 rows and 12 columns. The columns are the following: user\_id, session\_id, timestamp, step, action\_type, reference, platform, city, device, current\_filters, impressions, and prices.

# Preprocessing

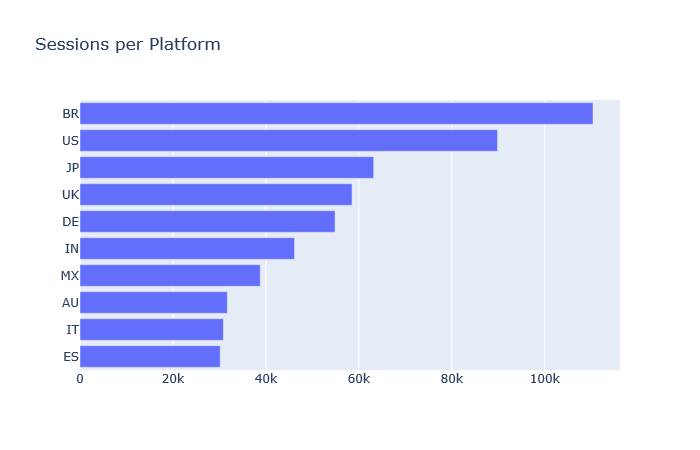
As part of the data cleaning process, I implemented the following steps:

* Feature engineered session duration and click out events
  + Duration was calculated by the difference in time (UNIX) between the first and last sessions for each user ID and then converted to seconds
  + Click out was determined if there was a click in any of the sessions for a particular user ID and if there was one or more, then it was included as True.
* Removed the following features: timestamp, step, reference, city, current\_filters, impressions and prices.
  + This removed missing values
* Used one hot encoding to transform categorical variables to numeric ones

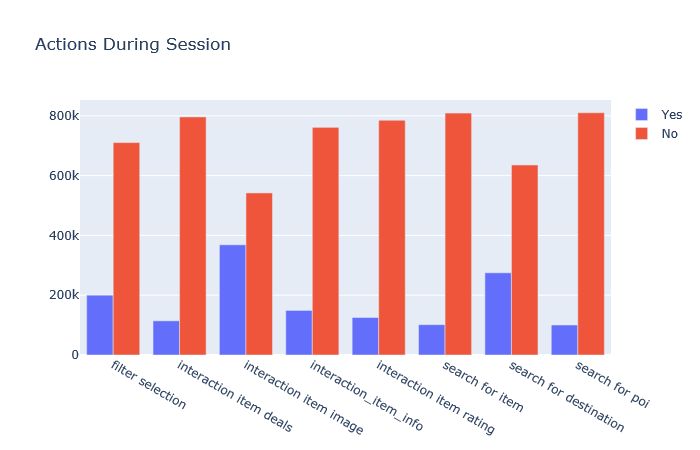
After completing the preprocessing, the dataset has 910,697 rows and 70 columns of which 655,204 are click and 255,452 are non-click events. This shows that the target variable was imbalanced. Given this, I down sampled the majority class resulting in 255,452 click and non-click events respectively. This ensured a balanced dataset as well as reduced the size of the dataset which was favorable for memory constraints.

# Exploratory Data Analysis

Before developing and testing models, I explored the data to get a better sense of its makeup and if there are any correlations between the variables. In this first graph, I wanted to look at where the customers are located. To do this I plotted the number of sessions by platform. Interestingly the U.S. does not have the most sessions on the platform, but rather Brazil does with 110,380 sessions. This is followed by the U.S. with 89,835 and Japan with 63,188 sessions.

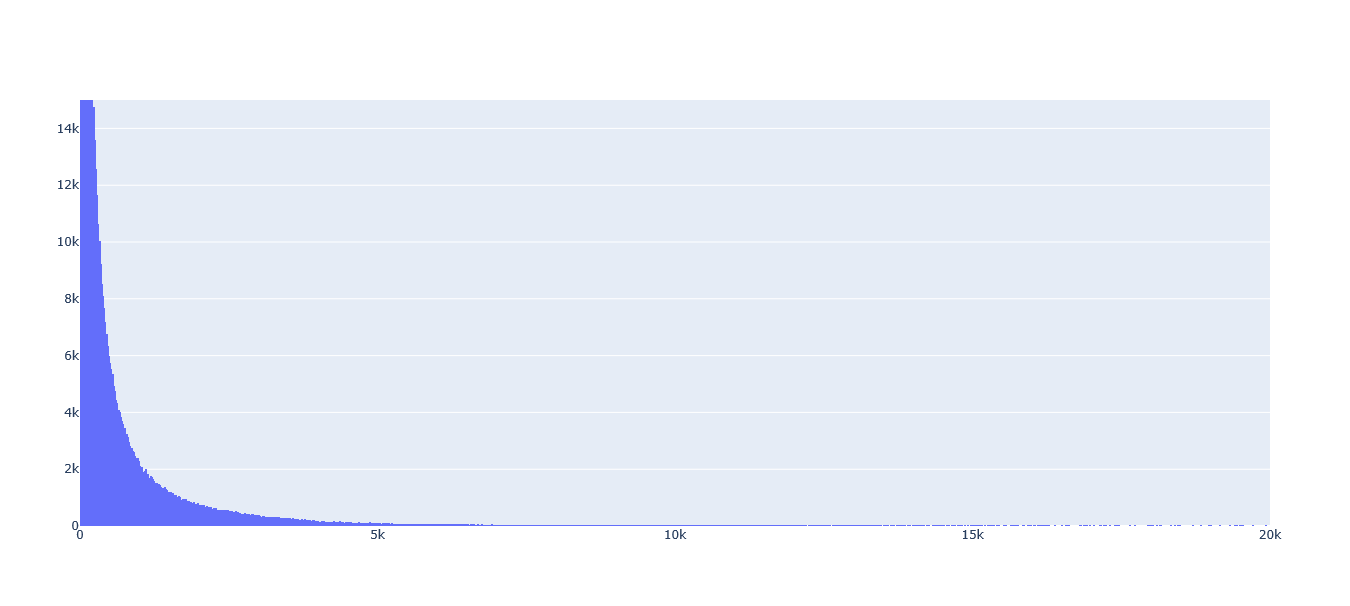


Next, I wanted to understand what actions were taking place specifically with respect to whether a user ended up clicking on an item or not. As we can see from the actions that were performed during session the most popular one was the interaction item image and the search for destination.

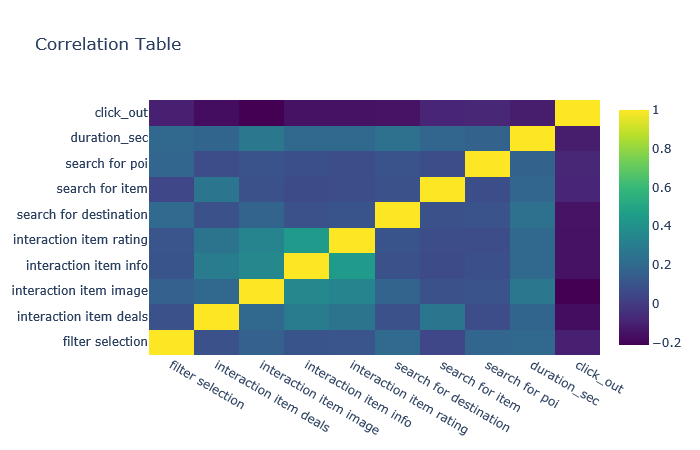


This is useful information because it shows which components of the website a user interacted with. The table shows that interacting with images had the highest click rate.

The histogram below shows the number of users and the duration of time spent on a given session. The vast majority of users spent 2500 seconds or less in a given session, which is about 40 minutes.

****

Finally I created a heatmap to visualize what, if any, correlations there are within the dataset. Below is the plotted correlation table. The stronger the correlation, the closer the color is to bright yellow. No correlation at all is dark purple.



Based on the correlation table above, we can see that there are some small variations in correlation up to about 50%. These are the strongest relationships we see which are between the interaction of an item rating and item information. This makes sense given that a rating may propel a user to look at the item’s information or vice versa. But on the whole, there is not a significant correlation between any two types of interaction, duration or click out.

# Modeling

After completing the preprocessing and EDA, the dataset has 910,697 rows and 70 columns of which 655,204 are click and 255,452 are non-click events. This shows that the target variable was imbalanced. Given this, I down sampled the majority class resulting in 255,452 click and non-click events respectively. This ensured a balanced dataset as well as reduced the size of the dataset which was favorable for memory constraints.

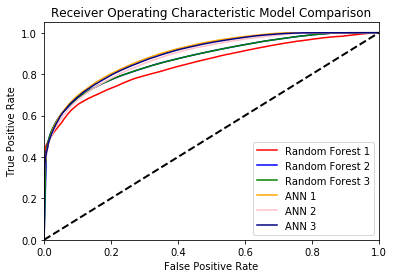
Used two different algorithms tuning hyperparameters for each of them to develop a total of six models:

Random Forest (n\_estimators = [100, 200, 400]), (max depth = [10, 100, 200])  
 ANN (solver = ‘adam’, activation = ‘relu’, hidden layer sizes = (50,100), (100,100), (100, 100, 50))

Results of the models are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Parameters | Precision | Recall | F1 Score |
| Random Forest (1) | N\_est 100, MD 10 | 0.723 | 0.639 | 0.654 |
| Random Forest (2) | N\_est 200, MD 100 | 0.750 | 0.738 | 0.743 |
| Random Forest (3) | N\_est 400, MD 200 | 0.750 | 0.737 | 0.743 |
| ANN (1) | 2 HL (50,100) | 0.806 | 0.759 | 0.776 |
| ANN (2) | 2 HL (100,100) | 0.786 | 0.749 | 0.763 |
| ANN (3) | 3 HL (100,100,50) | 0.801 | 0.752 | 0.770 |

The second Random Forest Model and first ANN models performed the best. When the results are plotted below on the ROC Curve, it is confirmed that the first ANN model with two hidden layers of 50 and 100 nodes performs the strongest.



# Conclusion

Based on the results the first ANN model with two hidden layers the first having 50 nodes and the second having 100 nodes performed the best, albeit slightly better than two other models with an F1 score of .776. Compared with a baseline 50% probability of predicting a click, this model performs significantly better.

While these results are compelling, there are additional areas of research that would contribute further. First in terms of model development, it would be interesting to add in KFold and feature selection to see how these processes would address outliers in the dataset and possible improve results.

From a time standpoint, given the large size of the dataset, it was not possible for me to run gridsearch coded in a modular way to determine the best hyperparameters. It would be beneficial to look into Hadoop and use map/reduce to be able to more efficiently test the models.

Examining this problem in a holistic way, additional research could be done to explore what the specific preferences that were determined by a user clicking on an item such as price, location or amenity. And going one step further, how can those preferences be used to create a recommender system that would provide customized hotel listings for each user?