Maximilien Boucher ‘24
B.A. Behavioral Decision Sciences
Public Repository

Predicting Airbnb Booking Requests using User Activity

Introduction

Driven by a keen interest in studying human behavior and its impact on product or service quality, I embarked on a project focused on Airbnb user data. The goal was to develop a classification algorithm predicting whether a user would send a booking request, considering factors such as session length. Airbnb, an online marketplace connecting hosts and travelers, provided the dataset from Kaggle, encompassing 7,756 user sessions across 630 unique visitors. Key information included visitor and session IDs, session count, device details, and timestamps for login, logout, messaging, searching, and booking requests.

Exploratory Data Analysis

During the exploratory data analysis, a pivotal variable for the analysis, "session length," was created using the respective logging on and logging off timestamps, capturing the duration users spent on the platform—an influential factor for predicting booking requests. The analysis delved into activity breakdowns, revealing a highly imbalanced dataset, with only 1.9% of sessions involving booking requests. Additionally, 15.9% featured searches, and 16.5% included message sending.

Then, I explored the impact of device and application usage (dim_device_app_combo) on user behavior. The assumption was that these factors not only influenced session duration but also affected engagement, such as messaging and booking requests. The barplot below illustrates that users with iPads and Desktops tended to spend more time on the platform, while those with Android or iPhones had shorter sessions. This device/application analysis became a crucial aspect of the project, guiding the subsequent predictive modeling.

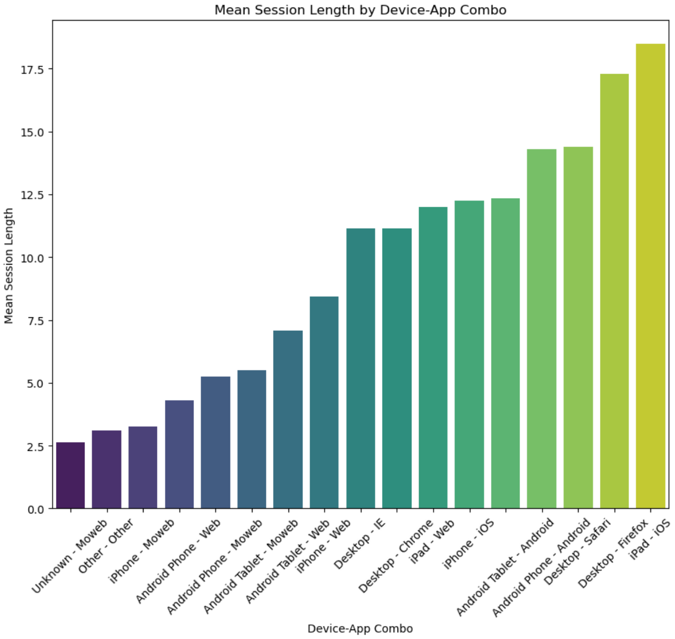

Figure 1: Mean session length depending on the the device/application combo used by user

I decided to follow down this line of thought by seeing if there was any relationship between the device used and our target variable (whether or not the user made a booking request). This analysis, as can be seen in Figure 2, revealed that users were much more likely to send a booking request if they were using a desktop. This is an important result to think about given that there were not a large number of bookings sent and over ⅔ of them were completed on a Desktop.

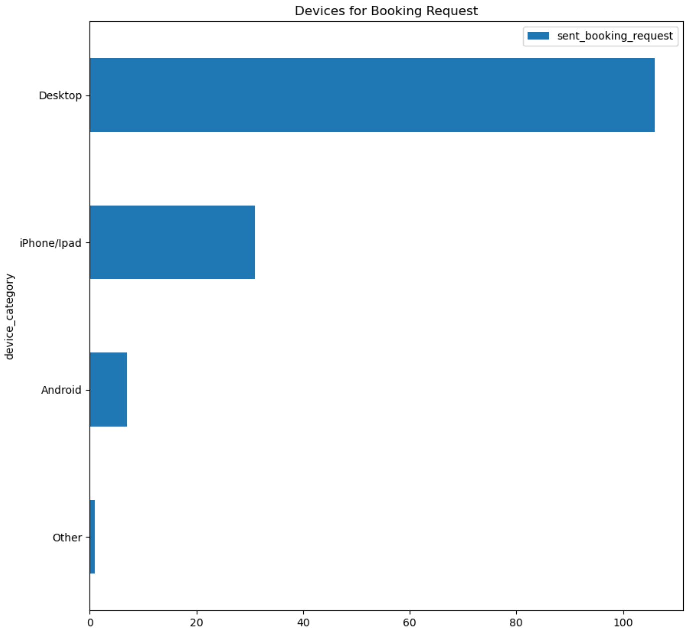

Figure 2: Count of sessions with booking request sent by device type

The next step in my analysis was to get a better understanding of all the numeric features as well as the possible relationships between them. In order to do this, I created a correlation matrix as can be seen below in Figure 3. I was surprised by the extremely low correlations across the board for all the variables with the highest positive relationship being between sent_booking_request and sent_message at 0.21 and the highest negative correlation being between did_search and dim_session_number at -0.22. These results made me realize that I may not have as good of an understanding of the variables as I thought. Indeed, one of the major limitations with the dataset I received is that there was no data dictionary so none of the variables were actually defined. This meant that I had to try and define them myself based on my understanding of how the Airbnb platform works which is a major limitation or weakness of the project.

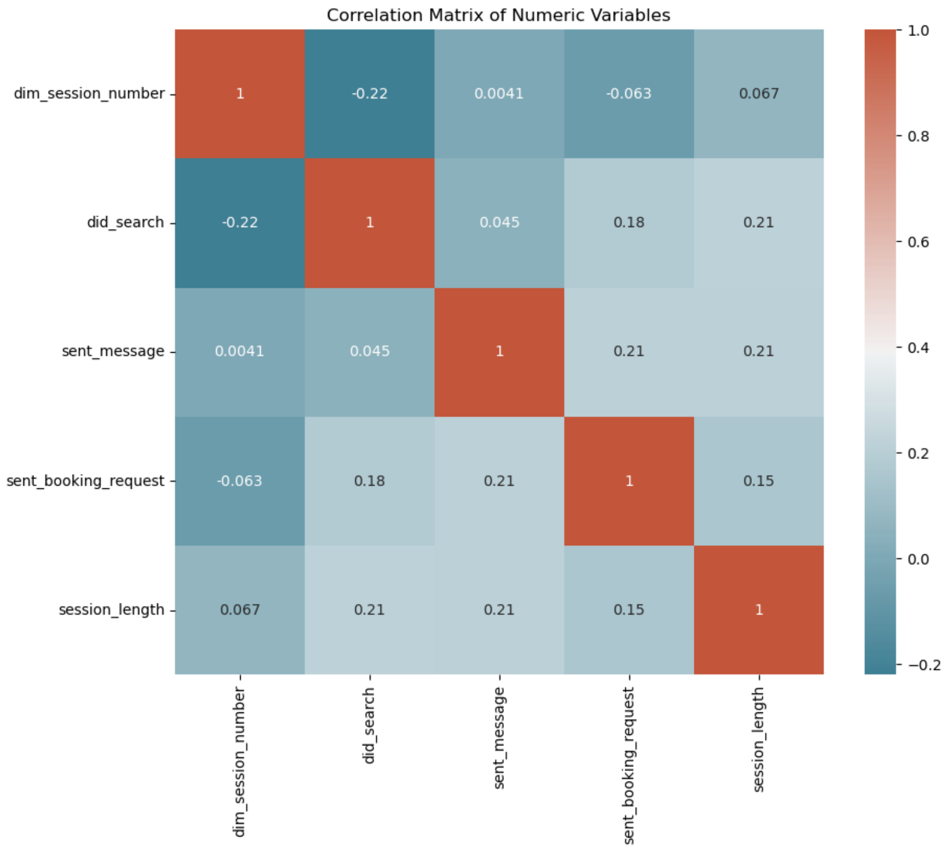

Figure 3: Correlation Matrix between all the numeric variables

The final component of this exploration consisted of mapping the lengths of sessions based on the activity being done by the user in that session which is one of the most insightful analyses done up to this point. As can be seen in Figure 4, out of all the three different groups of activity on the platform, those that searched and requested a booking spent the most amount of time on the platform and those that only searched spent the least amount of time. This makes sense intuitively when considering how someone that is just searching could be adopting the mentality of just browsing or looking around rather than seriously considering an airbnb to stay in.

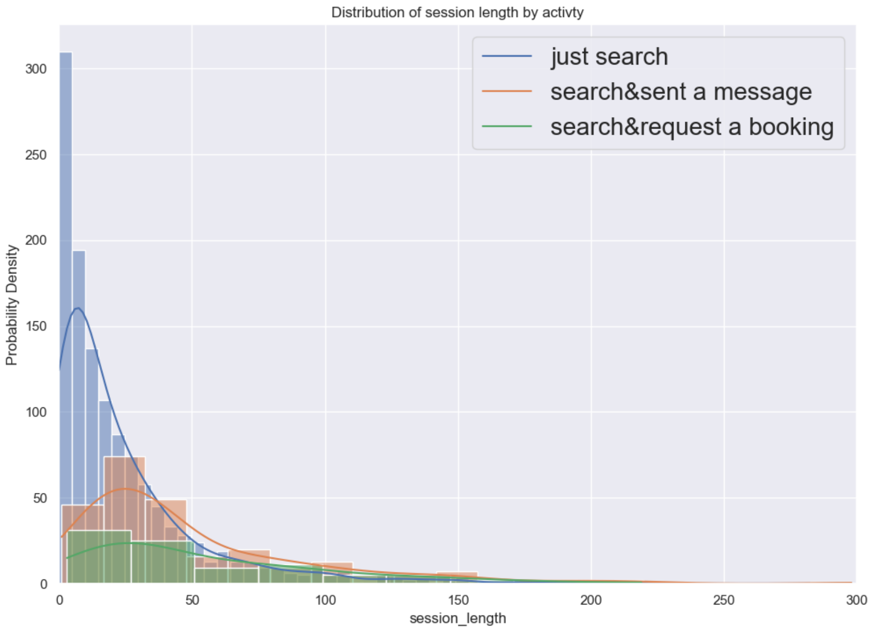

Figure 4: Distribution of session length based on activity

Methods

Splitting Strategy

As we are trying to predict if a new user will send a booking request in this question, our target variable is "sent_booking_request" which represents if someone sent the booking request (1) or did not (0). Since the data is obtained from different subjects with several samples per-subject, we need the model to be flexible enough to learn from highly specific features and avoid failing to generalize to new subjects. In effect, GroupKFold allows us to control overfitting by making sure that each subject is in a different testing fold, and the same group is not represented in both testing and training sets. Additionally, because of the imbalance in the data (there are more samples for certain users than others), each fold will not be exactly the same size. GroupKFold is a variation of k-fold which ensures that the same group is not represented in both testing/validation and training sets. For example if the data is obtained from different subjects with several samples per-subject and if the model is flexible enough to learn from highly personal specific features it could fail to generalize to new subjects. GroupKFold makes it possible to detect this kind of overfitting situations.

Data Preprocessing

The preprocessing stage involves the transformation of input features to facilitate the subsequent analysis. Three types of features, namely 'dim_device_app_combo,' 'dim_session_number,' and 'session_length,' undergo specific encoding techniques. For 'dim_device_app_combo,' a categorical variable with no inherent order, the OneHotEncoder is applied to create binary columns for each category. 'dim_session_number,' a continuous variable within a reasonable range, is normalized using MinMaxScaler to scale the values between 0 and 1. As for 'session_length,' a continuous feature with a tailed distribution, StandardScaler is employed for normalization, ensuring that the data conforms to a standard normal distribution with a mean of 0 and standard deviation of 1. It is noteworthy that 'did_search, sent_message and sent_booking_request'' are already standardized within the (0,1) range. The overall preprocessing workflow is encapsulated in a ColumnTransformer, which efficiently applies these transformations to the respective feature sets.

ML Pipeline

The machine learning pipeline designed for this project employs a GroupKFold cross-validation strategy with 5 folds, focusing on optimizing F-beta scores (beta=0.5) for four distinct algorithms: Logistic Regression, Random Forest, Support Vector

Classifier (SVC), and K-Nearest Classifier. Parameter tuning is conducted via GridSearchCV to enhance the performance of each algorithm. Logistic Regression undergoes tuning for the penalty term (L2 regularization), regularization strength (C), and maximum iterations. For Random Forest, parameters such as the number of estimators and maximum tree depth are fine-tuned to balance model complexity and prevent overfitting. SVC parameters include the kernel coefficient (gamma) and regularization parameter (C) to optimize hyperplane separation. K-Nearest Classifier is tuned for the number of neighbors and the selection of the weight function (uniform or distance) to refine the algorithm's configuration.

The chosen evaluation metric is F-beta with a beta value of 0.5, giving more weight to precision. This decision is motivated by the goal of minimizing false positives in the context of imbalanced data. Specifically, the dataset exhibits a significant imbalance, making any booking requests that are incorrectly predicted as positive (false positives) highly impactful. The choice of beta=0.5 reflects a preference for models that prioritize precision, emphasizing the reduction of false positives and ensuring that instances predicted as booking requests are genuinely significant. This consideration aligns with the nature of the dataset and underscores the importance of accurately identifying instances of positive class. The pipeline accounts for uncertainties arising from data splitting and non-deterministic methods like Random Forest, providing a robust assessment of model performance tailored to the dataset's characteristics.

Results

The baseline F-beta (0.5) score was calculated as 0.0205, representing the performance of a simple model that randomly predicts the positive class based on the fraction of sent_booking_request = 1 in the dataset. By comparing the mean F-beta scores of the evaluated models to the baseline, we can gauge how many standard deviations above the baseline each model performs. Logistic Regression demonstrated a mean F-beta score of 0.0571 with a standard deviation of 0.0702, indicating a moderate improvement over the baseline. Random Forest exhibited a mean F-beta score of 0.1282 with a standard deviation of 0.0812, suggesting a more substantial enhancement above the baseline. SVC yielded a mean F-beta score of 0.0658 with a standard deviation of 0.0689, while K-Nearest Classifier achieved a mean F-beta score of 0.0624 with a standard deviation of 0.0413. In comparison to the baseline F-beta score of 0.0205, the models' performances are notably elevated, with Random Forest emerging as the most predictive model due to its highest mean F-beta score and substantial improvement above the baseline as can be seen in Figure 5 below.

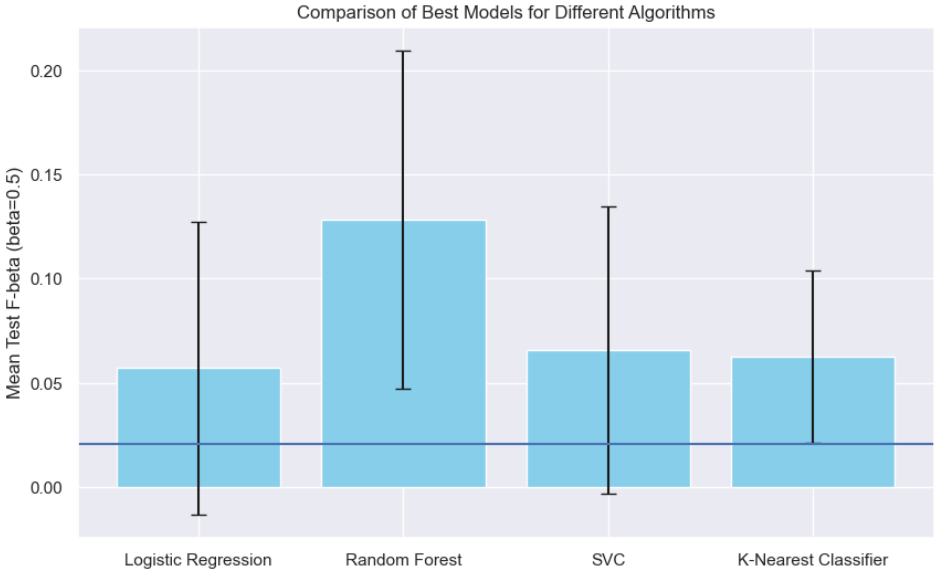

Figure 5: Comparison of Mean F-beta (0.5) for all the models and the baseline

When looking at the global feature importances for the best model, I used a permutation importance test to see how much the model's performance dropped when the values of a specific feature were shuffled. Looking at Figure 6 below it appears that the session length is the most important feature globally which is not surprising given both my intuition about how it would affect whether someone sends a booking request or not as well as the insights of its impact in the exploratory data analysis. This was followed by the session number as well as the device being desktop which are both not very surprising as well given their suggested influence through the exploratory data analysis.

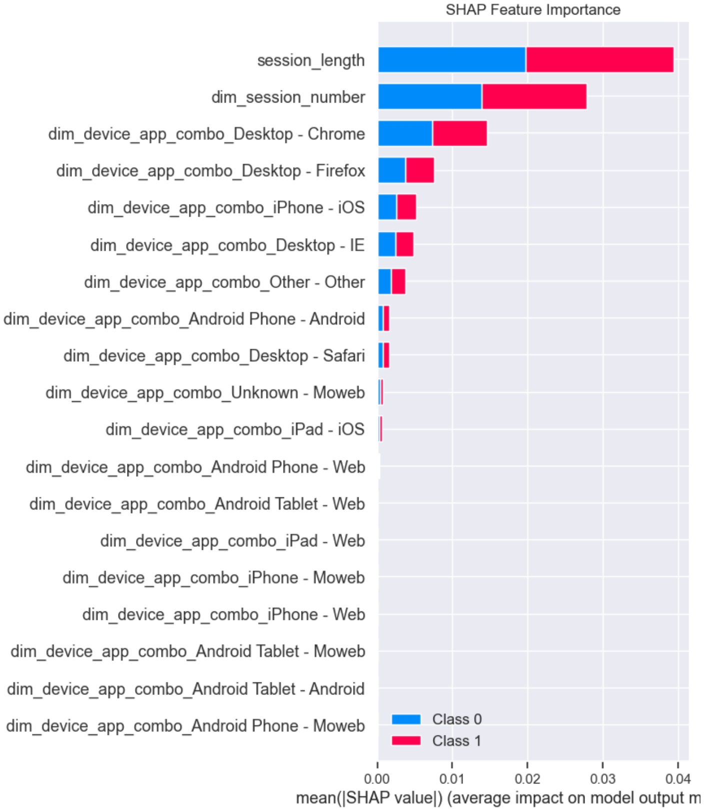

Figure 6: Permutation Importance Globally

As for the SHAP values for local feature importance, I looked at the data points at index 0, 500, and 1000. Below in Figure 7, which shows the prediction for the model at index 0, the model predicted that this data point reflected the activity of a user that would not send a booking request and that the features that contributed the most to this prediction were the dim_session_number and the device being a Desktop.

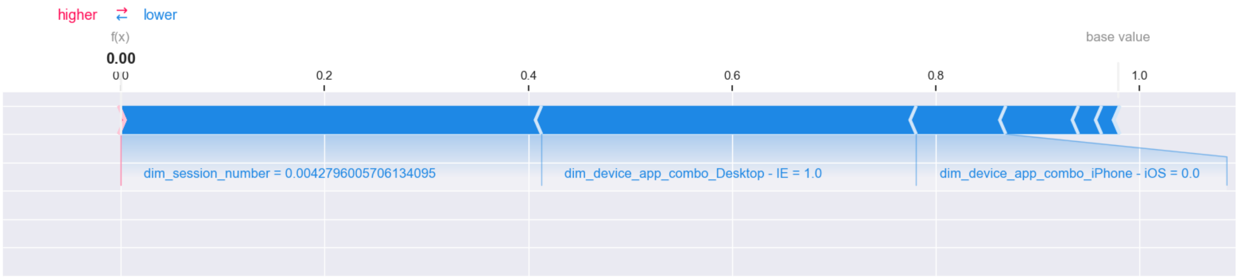

Figure 7: Prediction for data point at index 0

This finding is surprising in that I did not expect the fact that this user was using a Desktop to contribute so strongly to a prediction of not sending a booking based on what I saw in the EDA. As for the data point at index 500, shown in Figure 8, for which the model predicted that the user would send a booking request, it is not surprising to see that the session length was the strongest contributing factor to the prediction.

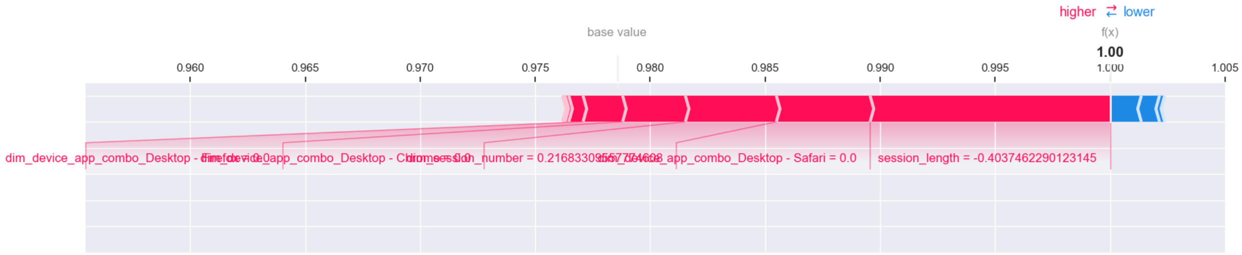

Figure 8: Prediction for data point at index 500

Outlook

To enhance the predictive power and interpretability of the model, several avenues for improvement can be explored. Firstly, obtaining better data, either through an increase in volume or more precisely defined data, could significantly enhance model performance. Additionally, a missed opportunity lies in feature engineering, specifically considering the frequency of user visits, which could provide valuable insights into user behavior. Expanding the repertoire of machine learning models by testing others like XGBoost and Naive Bayes Classifier may uncover more suitable algorithms for the given data characteristics. To further refine the model, an exploration of additional data dimensions or variables related to user behavior could be beneficial. Incorporating these strategies would contribute to a more robust and nuanced predictive model.

References

https://www.kaggle.com/datasets/heeraldedhia/airbnb-user-pathways/data

https://medium.com/towards-data-science/baseline-models-your-guide-for-model-building-1ec3aa244b8d

https://medium.com/@douglaspsteen/beyond-the-f-1-score-a-look-at-the-f-beta-score-3743ac2ef6e3#:~:text=The%20F%2Dbeta%20score%20calculation,recall%20using%20the%20beta%20parameter.

https://spotintelligence.com/2023/05/08/f1-score/#:~:text=However%2C%20as%20a%20general%20rule,false%20positives%20and%20false%20negatives.