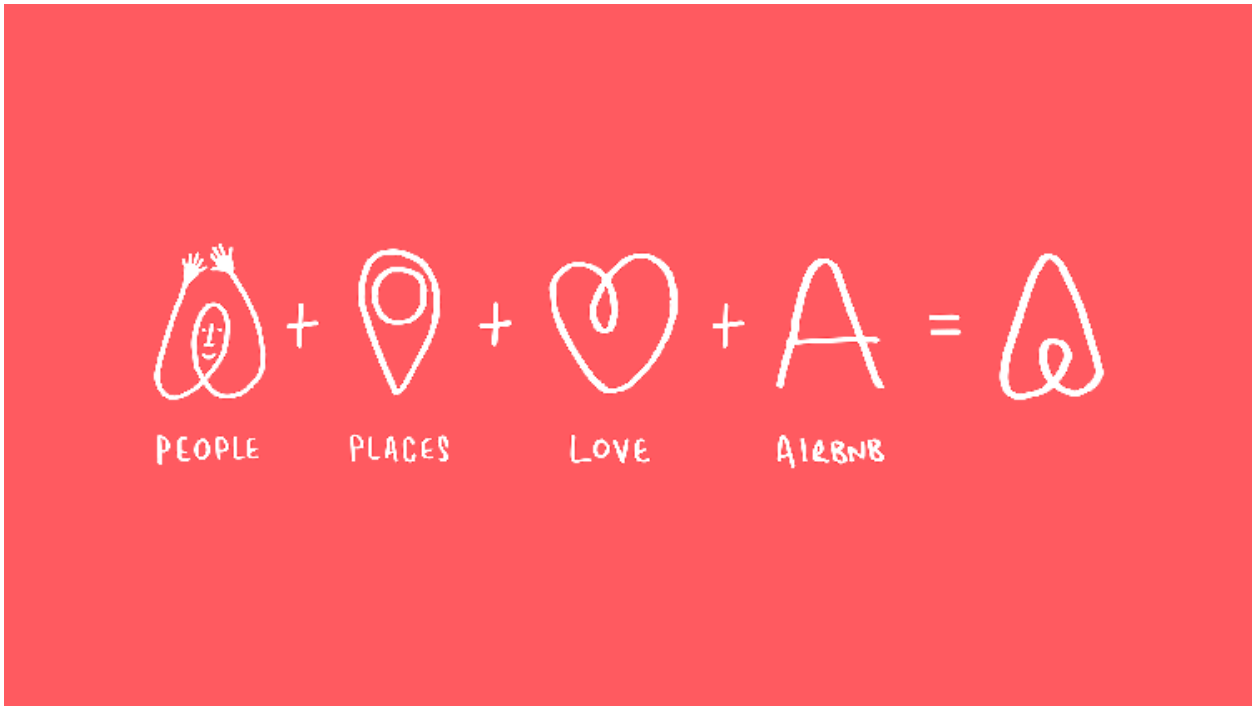

Airbnb Top Listings

Predicting the popularity of a Airbnb listing

06.08.2021

Valentina Rizzati
Data Scientist
Airbnb, Host Team

Opportunity

I am a Data Scientist for the Host team at Airbnb and I've recently been approached by a Product Manager who intends to enhance the set of Host tools with a dashboard visualizing, amongst other things, two categories of listings: top listing and not top listing. This is a binary classification problem that I intend to solve with a classification methodology (e.g. kNN or logistic regression).

As aligned with my business counterparts, the first prototype of this dashboard will be based on the Airbnb listings in NYC.

Impact Hypothesis

If hosts have more visibility on the expected performance of their listing, they will be able to optimize the listing's most critical features and work towards improving their performance and achieving the *Top Listing* status. The more listings and guests' experiences are optimized, the higher retention will be on the Airbnb platform.

Data

I have collected data about around 36,000 Airbnb listings in NYC from Inside Airbnb.

TARGET: For the purpose of this analysis, my target variable - *Top Listing* - is defined as a listing with *Avail_365* lower than or equal to than the median of *Avail_365*.

FEATURES: different features are included in different versions of my model, as I am still working on selecting the optimal model. The current winning version of the model (F1 beta = 0.61) is a logistic regression featuring the following 11 features:

- Number of host listings | numerical discrete
- Length of *Host About* description | numerical discrete
- Number of Rooms | numerical discrete
- Number of Reviews | numerical discrete
- Length of listing Description | numerical discrete
- Count of amenities | numerical discrete
- Host Since | converted to numerical discrete
- Instant Bookable | dummy
- Host in the US | dummy
- Brooklyn | dummy
- Hotel Room | dummy

Algorithm

DATA CLEANING & EDA

Data cleaning was a critical step in my process. In fact, by noticing that the median of *Availability_365* (i.e. the number of days the listing is available in the next 365 days and cutoff point for *Top Listing*) was only 56, I was able to identify a large amount of inactive listings, presenting zeros in multiple columns. As I intend for my model to be only focused on active listing (inactive listings cannot be booked on Airbnb, hence I do not intend to take them into account for the prediction) I removed inactive listings in line with a couple of assumptions. Hence, the amount of rows my baseline dataset was composed of was around 26,000 and the new median of *Availability_365* increased to 156, a level that is definitely more aligned to what I would expect from my domain knowledge.

OPTIMIZATION METRIC: F1 BETA

As the metric to optimize my model for, I chose F1 beta because precision and recall are equally important for the purpose of my analysis. I chose a beta of 0.35 because I intend to weight precision more (i.e. I want to make sure we don't tell some hosts that they have a Top Listing while they do not, as this error would have significant behavioral implications from a host perspective).

BASELINING: LOGISTIC REGRESSION vs. kNN

Once the data was cleaned, I started baselining my model with both Logistic Regression and kNN. In this step, I also tuned the hyperparameter k to maximize the F1 beta and found the optimal k (in a range from 1 to 40) to be equal to 20.

FEATURE ENGINEERING & MODEL EXPANSION

Then, I conducted feature engineering and created new features like *num_rooms* (number of rooms), *price_pp* (price per person), *content_len* (content length).

By iterating on different combinations of features, I ended up running 22 versions of my model for both Logistic Regression and kNN (i.e. 44 distinct model versions).

TESTING OUTLIER REMOVAL & REMODELING

Then, I tested the elimination of outliers in accordance with a logic that I deemed appropriate for the business case and refined the dataset to around 20,000 rows. I then rerun the 44 model versions (Logistic Regression and kNN) and scored them on F1 beta.

Knowing that Logistic Regression is more sensitive to outliers, I don't notice any change in the metric of interest (F1 beta). Therefore, I have decided I will keep the outliers as part of my model.

Tools

To build the classification model I used pandas, numpy, scikit-learn.

For diagnostic visualizations within the Jupyter Notebook I used seaborn and matplotlib.

Finally, depending on time availability, I will create host-facing visualizations with Tableau or Flask.

Next Steps

RANDOM FOREST

I Intend to test a random forest model and then compare it with my Logistic Regression and kNN, mindful that model interpretability is a critical component of my deliverables.

CROSS VALIDATION

I will cross-validate 2-3 top versions of my model and choose the winning version.

HYPERPARAMETERS TUNING

I will tune the hyperparameters of the winner model to maximize F1 beta.

VISUALIZATION WITH FLASK OR TABLEAU

Depending on time availability, I will create an interactive visualization on Tableau (descriptive of current Top Listings) or Flask (allowing the user to insert values for.