Predict Airbnb Listings' Availability in San Francisco

In fulfillment of Springboard's Capstone Project I

Fan Dong

Executive Summary

- The project aims to build a model that predicts a given Airbnb listing's availability for a given time period.
- Data: Detailed information of 392 Airbnb listings in San Francisco, with their daily availability info from September 2015 to September 2017; properly processed
- Method and Result: 7 models tested. **XGBoost** has the highest accuracy of 95.85%.
- The model is a valuable tool for both guests (planning travel wisely) and hosts (knowing listing's value and planning ahead).
- Based on the result of the simulation case, there is not necessarily a linear relation between price and booking rate (for a given listing) in the short-term lodging sharing business.

Outline

- Motivation and Goal
- The Data
- The Model
- What Does Our Model Tell Us?
- A Real Life Application
- Directions worth exploring in the future

The Pioneer of ‘Sharing Economy’, Airbnb provides a platform on which hosts rent out their own places for extra cash and guests find suitable accommodations for reasonable prices and different travel experiences.

Listings are the key products that users on the two sides of the market are sharing. We are interested in evaluating their values.

While price is readily available, the quantity - number of nights a place has been booked is not.

Thus, in this project, we aims to **build a model that predicts a given listing’s availability for a given time period.**

Data

Detailed information of 392 Airbnb listings in San Francisco, with their daily availability info from September 2015 to September 2017.

Obtained from: http://insideairbnb.com/get-the-data.html

The Data Wrangle Process

We went through the following steps to turn the original data into a tidy form that is readable by the models:

- Truncate and combine the raw files (for each month, there are one file with listing info and one with calendar availability info, more than 40 files in total) into one dataset
- Check for anomalies in the dataset and fix them
    - E.g.: studio has 0 bedroom; change it to 1
- Convert all variables to appropriate format
    - make numeric variables truly numeric
    - turn string variables into dummies

The Data Wrangle Process

- Make full use of 'Missing Values' by creating informative variables
  - 'Missing Value' dummies for those with too many NAs
  - Create variable that indicates listing description's text length
- Drop uninformative variables and rows
  - E.g., Amenities that are only provided by a limited amount of listings
- Create a series of date-related features
  - Month, year, week dummies
  - Weekend, holiday dummies

The ‘Ready-for-Modeling’ Data

286558 Observations, 123 Features

Here’s how the data look like (the first row):

accommodates	bathrooms	bedrooms	beds	calculated_host_listings_count	extra_people	guests_included	host_has_profile_pic	host_identity_verified
4	2.0	3.0	3.0	16	50.0	4	True	False

has_host	summary_len	holiday	host_for	book_month	book_year	book_weekday	weekend	book_week	target
0	192	False	64	9	2015	1	False	36	1

Some Key Features

Price

# of ppl accommodated

# of bedrooms/bathrooms/beds

Property/bed type

Neighborhood

Availability of key amenities

Booking date

Maximum/minimum nights

Length of being a host

Host’s listing counts

Text length of the listing’s summary

Number of Reviews

The Model

Method: Binary Classification Prediction that returns the probability of whether a listing will be available on a given day; we say the listing is available if the probability is greater than 0.5, otherwise it is unavailable.

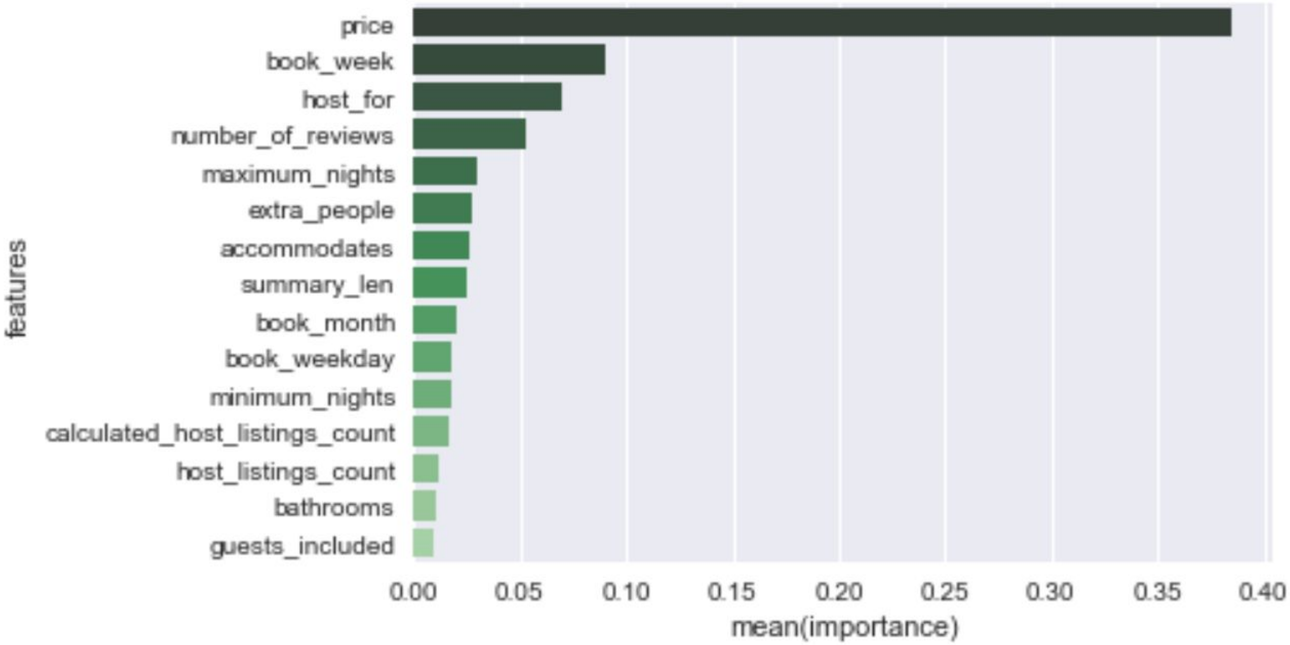
The testing results of a series of classification models are listed on the right. 4-fold cross-validation is used in all cases.

XGBoost appears to be the clear winner.

Model	Accuracy
LASSO	70.45%
SGD	60.61%
AdaBoost	77.32%
Neural Network	78.02%
Random Forest	79.23%
Gradient Boosting	93.17%
XGBoost	95.85%

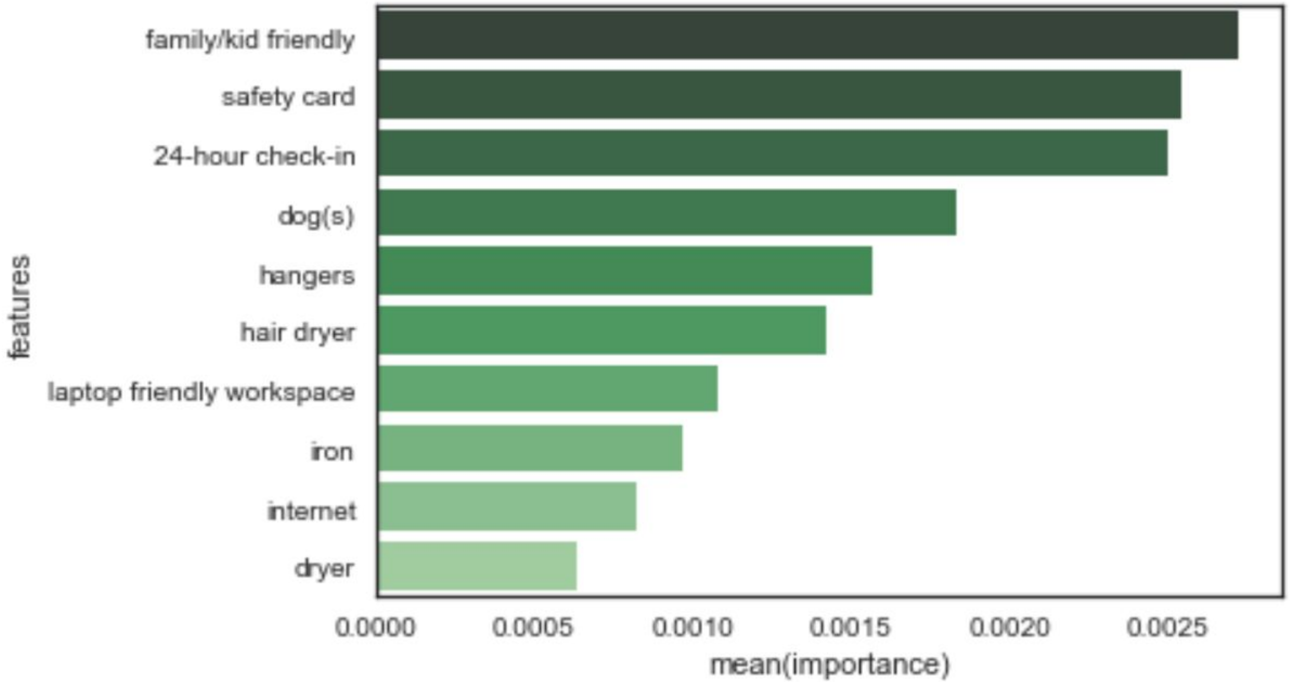
What Does Our Model Tell Us?

Features that play an important role in booking-decision-making process:



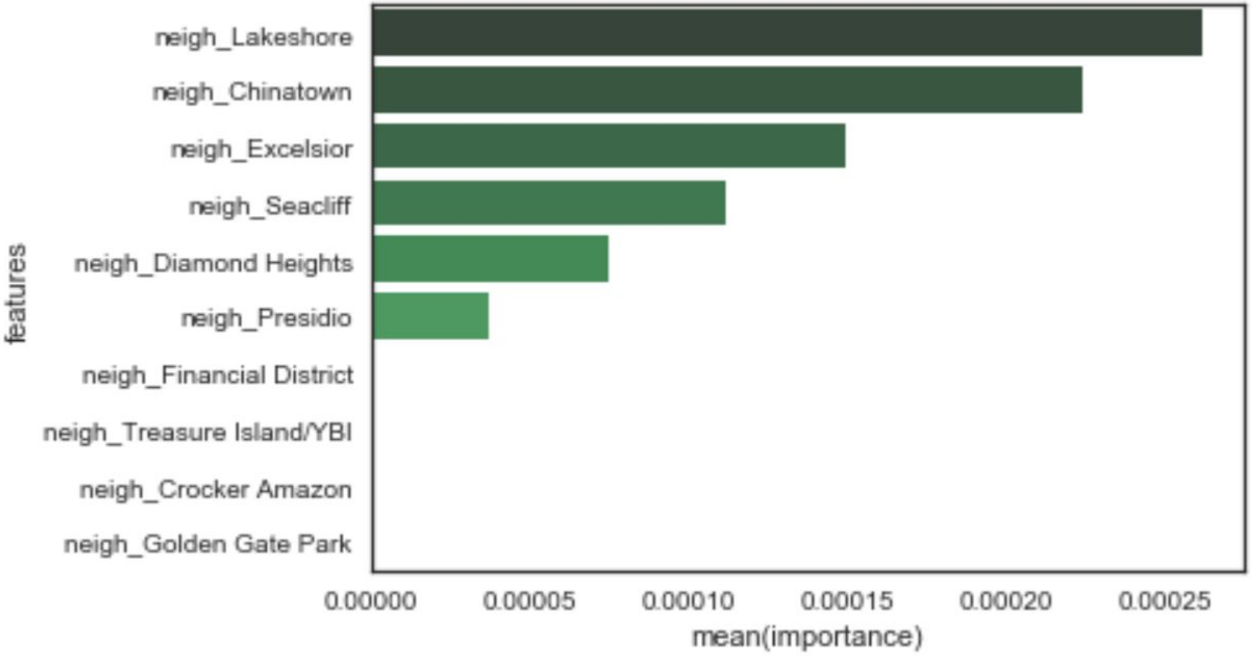
What Does Our Model Tell Us?

Amenities that help your place sell fast:



What Does Our Model Tell Us?

The neighborhoods that matter:



A Real Life Application

ENTIRE HOUSE

Spacious, clean, quiet suite

San Francisco

LayKoon

2 guests · 1 bedroom · 1 bed · 1 bath

Business travelers, welcome! Undisturbed privacy.
Separate entrance, access to lovely garden and deck. Close to SFO airport, public transit and freeways. Direct Muni light rail to Kaiser and UCSF Mission Bay, as well as AT&T stadium. 2.3 miles from Cow Palace.
Free street parking near listing if you find a spot. WiFi included. Nice home away from home for business travelers on longer stays.
Registration #STR- (Phone number hidden by Airbnb)

Read more about the space

Contact host

Amenities

Kitchen   TV

$115 per night
★★★★★ 74

Dates
06/25/2018 → 06/30/2018

Guests
1 guest

$115 x 5 nights	$575
Cleaning fee ?	$75
Service fee ?	$83
Occupancy Taxes ?	$102
Total	$835

Request to Book

You won't be charged yet

As a guest...

We want to book this "Spacious, clean, quiet suite" from **June 25, 2019 to June 30, 2019** for a work trip. The place is not available for booking yet. **If we come back in early June 2019, will this place be available for the period of time we want?**

And the model says...

```
real_case_pred = clf.predict(real_case)
```

```
real_case_pred
```

```
array([0, 0, 0, 0, 0, 0], dtype=int8)
```

Six 0's!

All six days of June 25 2019 to June 30 2019 will be booked!

As a host...

Suppose LayKoon, the host of this listing, is using our model. She wants the model to answer the following questions:

- How's my place's booking rate for the summer (June 3 2019 ~ September 1 2019)?
- How will the booking rate/total revenue change if I adjust the price?

For the first question...

The model gives the following result:

```
real_whole_pred = clf.predict(real_case_whole)
```

```
real_whole_pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 0, 0], dtype=int8)
```

We can translate it to a more human-readable format...

2019 June

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
27	28	29	30	31	01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

2019 July

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
01	02	03	04	05	06	07
08	09	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	01	02	03	04

2019 August

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
29	30	31	01	02	03	04
05	06	07	08	09	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	01

Unavailable: dates marked in gray

Available: otherwise

Booking rate = 59.3%

Total projected revenue = $ 6310

For the second question...

Let's try a couple of scenarios as an example:

	Booking Rate	Total Revenue
Decrease price by $10	29.7%	$2987
Original	59.3%	$6310
Increase price by 25%	46.2%	$6190
Increase price by 40%	47.3%	$7155

Directions worth exploring in the future

- Get more data and train the model for a longer period of time / for other cities
- Dynamic prediction (exactly on what day has a booking/cancellation taken place)
- Integrate spatial analysis into the model