Predicting Hotel Bookings

Akaash Mungale
August 2017

Background

- We are conducting analysis for a popular US based hotel booking website
- We would like to understand user booking behavior to help target advertisement optimize user experience while maximizing revenue

The Problem

We will approach this problem by answering three questions.

Given a user's profile and browsing can we predict:

1. **if they will make a booking**
2. **if they will make a booking domestically (US) or abroad**
3. **if they book abroad, which country they will make their booking in?**

The Data

We are working with two data sets:

1. User signups and bookings
   - anonymized data including fields a user enters when signing up for the website as well as country of first booking
   - fields include user id, age, gender, first booking country (if any), and several others

```
len(users.columns),len(users)
```

```
(16, 213451)
```

```
users.columns
```

```
Index(['id', 'date_account_created', 'timestamp_first_active',
       'date_first_booking', 'gender', 'age', 'signup_method', 'signup_flow',
       'language', 'affiliate_channel', 'affiliate_provider',
       'first_affiliate_tracked', 'signup_app', 'first_device_type',
       'first_browser', 'country_destination'],
      dtype='object')
```

2. User browsing sessions
   - incomplete data set capturing user interactions with the website
   - fields include type of device, interaction type, time of interaction

	user_id	action	action_type	action_detail	device_type	secs_elapsed
0	d1mm9tcy42	lookup	NaN	NaN	Windows Desktop	319.0
1	d1mm9tcy42	search_results	click	view_search_results	Windows Desktop	67753.0

Data Wrangling

We will use Pandas to work with data. Some important steps taken to clean the data are outlined below with examples:

1. Convert dates and times to useable formats using Pandas time series functionalities

```
#Convert dates to useable formats
users['date_account_created'] = users['date_account_created'].apply(lambda x: pd.tslib.Timestamp(x))
```

```
users['date_first_booking'] = users['date_first_booking'].apply(lambda x: pd.tslib.Timestamp(x))
```

```
users['date_first_active'] = users['timestamp_first_active'].apply(lambda x: pd.tslib.Timestamp(x))
```

2. Clean incorrectly entered data from user signups, like users who entered their birth year instead of age

```
# Clean Age
users['created_year'] = users['date_account_created'].apply(lambda x: x.year)
users["age"] = users.apply(lambda r: (r["created_year"] - r["age"]) if r["age"] > 1000 else r["age"], axis=1)
# Remove outliers that we consider 'fake'
users['age'] = users['age'].apply(lambda x: np.nan if x>90 else x)
users['age'] = users['age'].apply(lambda x: np.nan if x<18 else x)
```

3. Fill NA values with -1 to improve classification

```
# Fill -1 for classifiers. We already have an NA count column
model_data = train_set.fillna(-1)
```

Feature Engineering

To create the final data set, we engineer several features. Examples:

1. Using Pandas time series functionalities to create variables regarding user signup time

```
# Add features regarding date of account creation and date of first active use
users['created_day'] = pd.DatetimeIndex(users.date_account_created).day
users['active_day'] = pd.DatetimeIndex(users.date_first_active).day
users['created_weekday'] = pd.DatetimeIndex(users.date_account_created).weekday
users['active_weekday'] = pd.DatetimeIndex(users.date_first_active).weekday
```

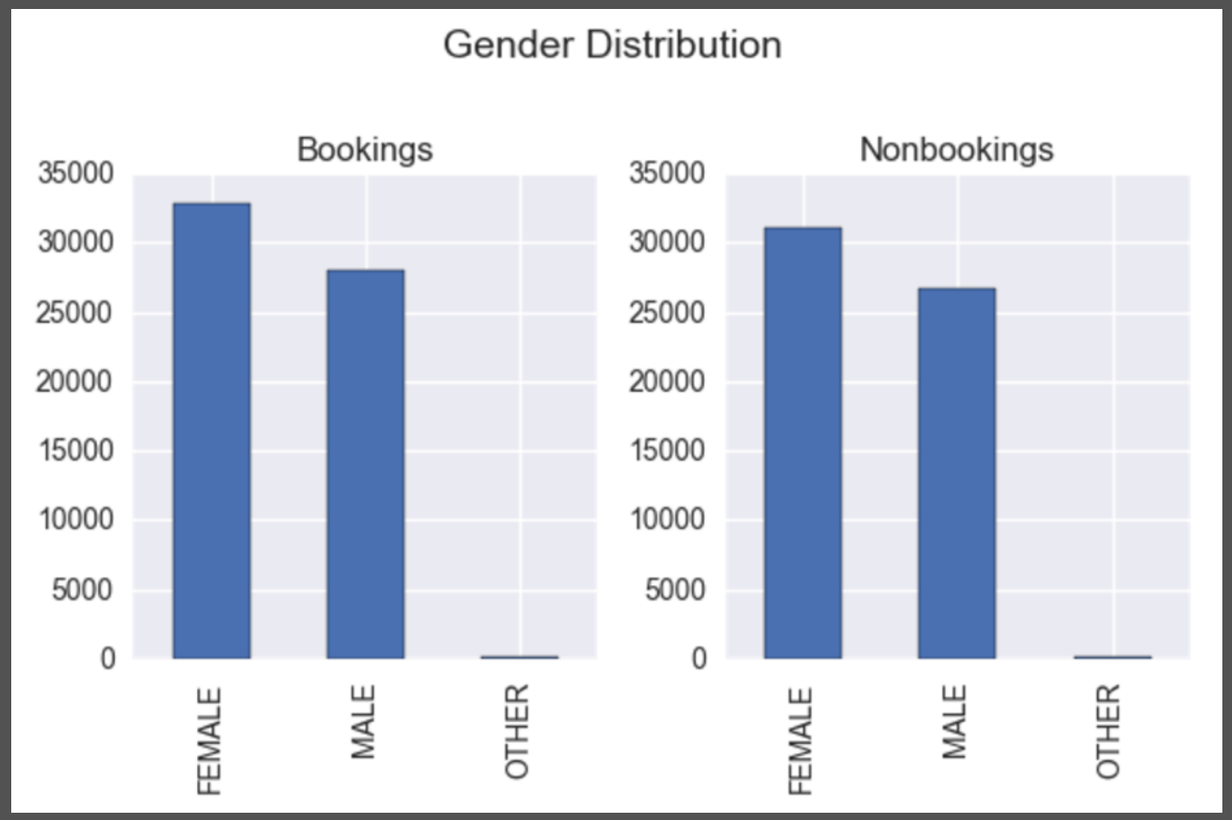
2. User's generational group (millennial, etc)

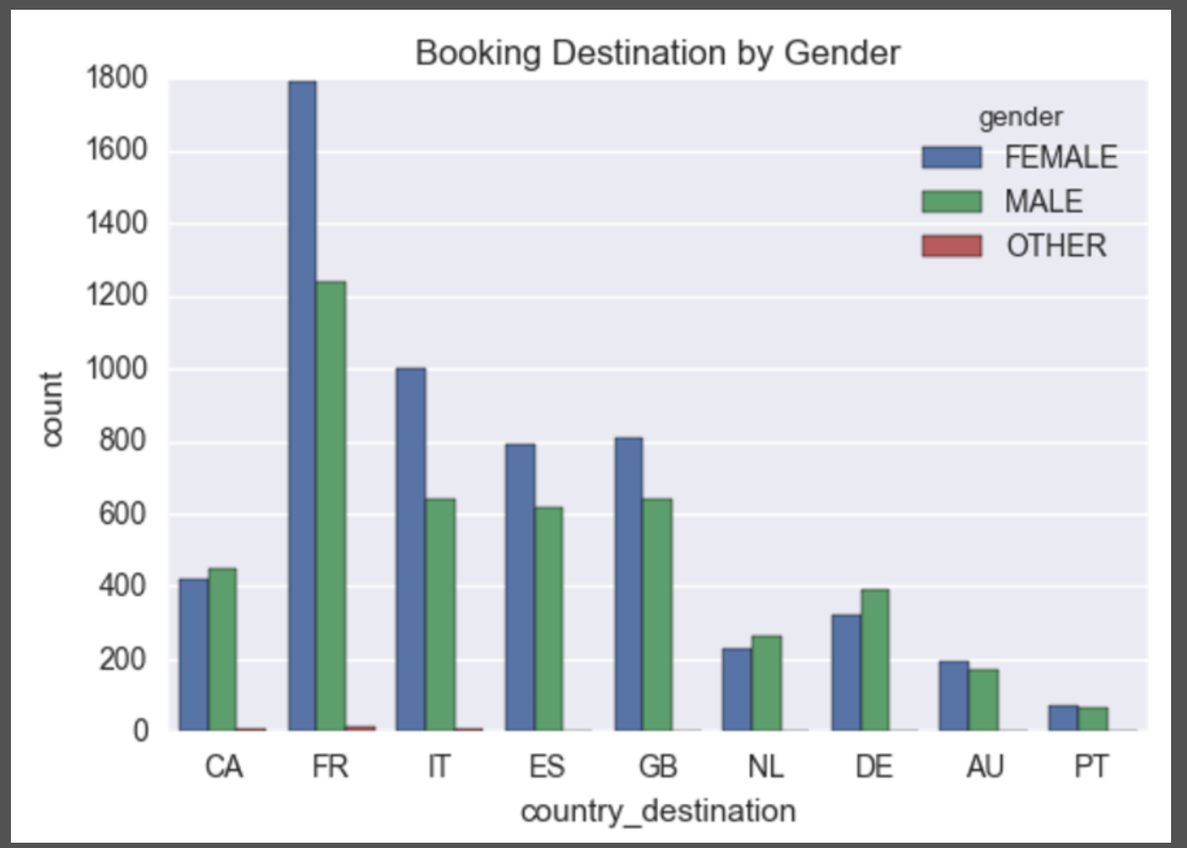
```
# Group users by their generation
users['ageGroup'] = np.nan
users['ageGroup'][users['birthYear'] <= 1996] = "Millenial"
users['ageGroup'][users['birthYear'] <= 1980] = "Generation X"
users['ageGroup'][users['birthYear'] <= 1960] = "Baby Boomers"
users['ageGroup'][users['birthYear'] <= 1942] = "Silent Generation"
users['ageGroup'][users['birthYear'] <= 1924] = "GI Generation"
```

3. Different aggregations of interaction time with the website

```
interaction_time_data = sessions.groupby(['id'])['secs_elapsed'].agg({'total_interact_time': np.sum,
                                                                      'median_interact_time': np.median,
                                                                      'mean_interact_time': np.mean,
                                                                      'min_interact_time': np.min,
                                                                      'max_interact_time': np.max,
                                                                      'var_interact_time': np.var})
```

Data Visualization - by Gender

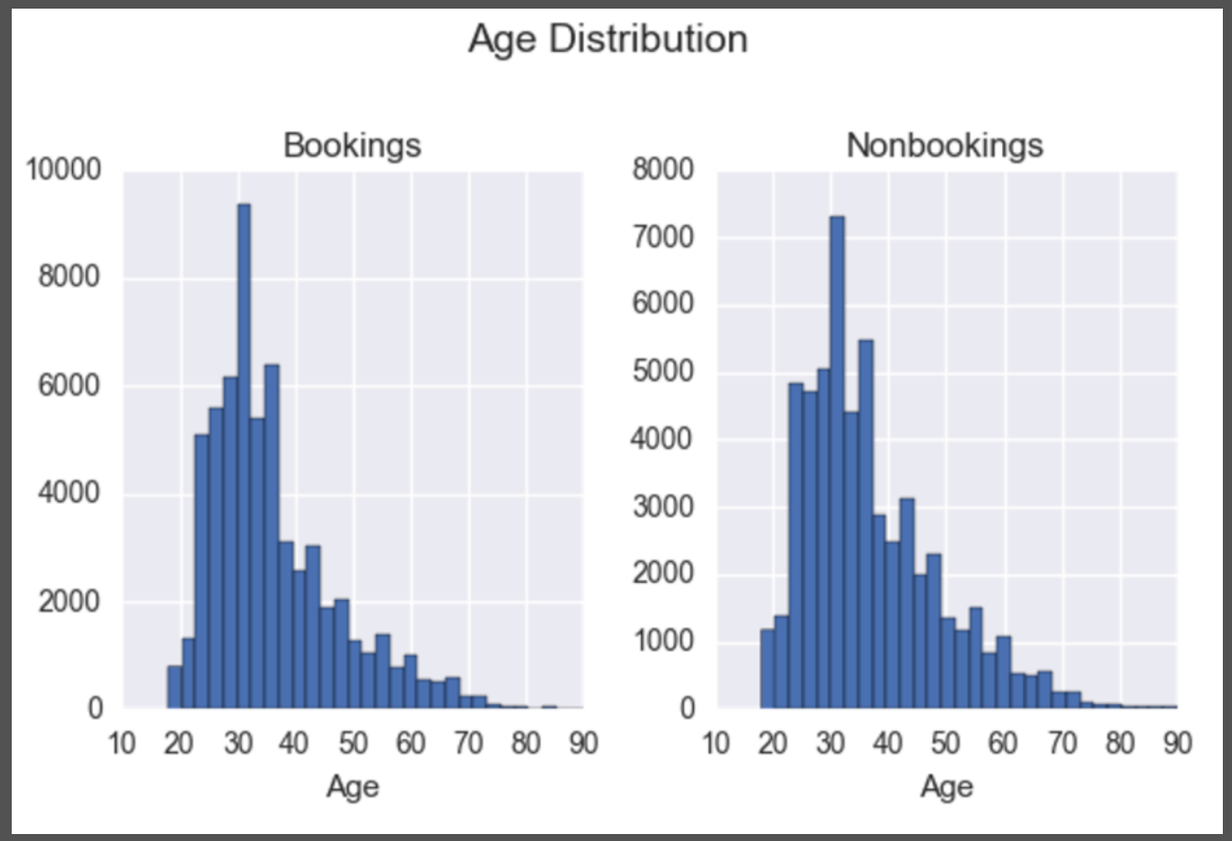


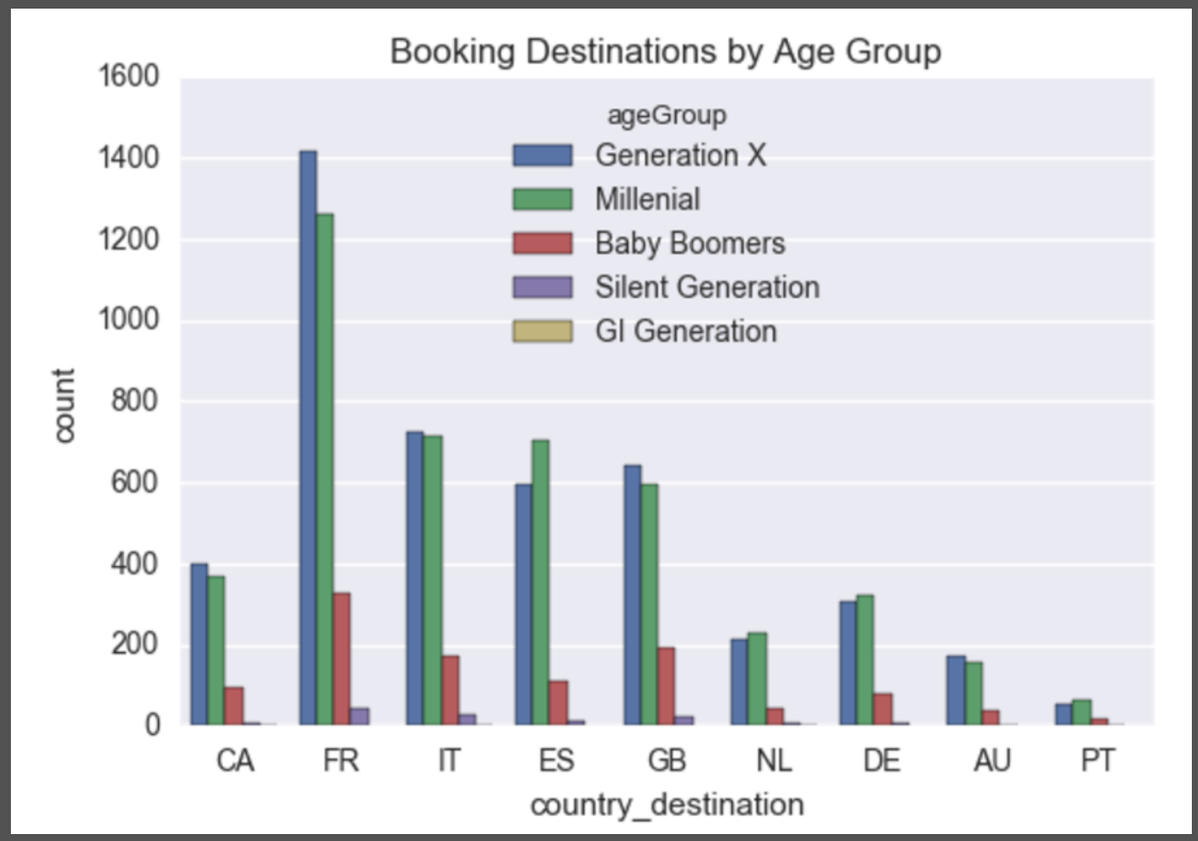


- More females signed up to the site than males. A higher proportion of actual bookings were also made by females

- 58% of signups didn’t end up booking. 29% first booked in the US. The booking distribution of the remaining 13% is shown above

Data Visualization - by Age





- Booking and non booking age distributions are right skewed. The site is most popular amongst 25-35 year olds.

- The highest number of bookings were by female Millennials (younger women), followed by Generation X males (middle aged men)

Will the user book a hotel?

Approach:

- Create an indicator variable for if a booking was made

```
bookings['madeBooking'] = bookings['country_destination']!='NDF'
```

- Fit a binomial regression model to the data (using scikit learn)
- Tune the model with gridsearch, determine significant predictors

Results:

- 78% precision (mean, over 5-fold cross validation)

Insights:

- Surprisingly, gender was not a significant predictor of whether someone will book
- Age was the most significant predictor, followed by foreign language preference

Will the user book in the US?

Approach:

- Create an indicator variable for if a booking was made, and if it was made in the US.
- Again, fit a binomial regression model to the data (using scikit learn)

Results:

- 78% precision (mean, over 5-fold cross validation)

Insights:

- The most significant predictor is now foreign language preference indicator - which is intuitive (if the preference is for English or not for English)

```
languageIndicator_us = pd.DataFrame(madeBookings['language']!='en')
```

- Prediction accuracy decreases with a more specific question, but remains fairly high

Where will the user book?

- This was the most challenging part of the predictions, and took several approaches. The most successful were:
  - Multinomial Regression (~63% Accuracy)
  - Random Forest (~68% Accuracy)
  - GBM (~65% Accuracy)
  - XGBoost (~72% Accuracy)

- Results: For the final model we used Stacked Generalization to ensemble our tuned Random Forest model and our XGBoost model, which ended up with a mean precision accuracy of 73% (mean over 5 fold cross validation)

	precision
0	0.70
1	0.79
2	0.74
3	0.71
4	0.77
5	0.69
6	0.70
7	0.73
avg / total	0.73

Summary and Insights

1. Using our binomial logistic regression model, we can predict with over 80% accuracy if someone will make a booking
2. Using our second binomial regression model, we can predict with 78% accuracy if a user will book in the US - the company can use this model to target specific promotions
3. Using our ensemble model constructed with Stacked Generalization (ensemble of Random Forest and XGBoost models) the company can make country specific recommendations to users, with 73% accuracy
4. In general, the best candidates for the website are Millennial women and middle aged men (30-40) who signed up directly through the website (as opposed to a social media referral), and those who indicate a non-English foreign language preference in their account.

Next Steps

If I had unlimited processing power, next steps I would take would include

- Combining our data set with economic and tourist data sets of the destination countries for additional feature engineering
- Further investigation of Stacked Generalization and additional permutations of stacked models
- Finer parameter tuning with XGBoost with a more extensive grid search

Citations

My full code: https://github.com/amungale/Hotel-Booking

Huge thank you to David Gasquez (https://github.com/davidgasquez/), whose feature engineering approach to this data set greatly inspired my own.

Images:
omnihotels.com
e-architect.co.uk/america/american-hotel-buildings
snowtravel.com.ua
visitlasvegas.com
images.trvl-media.com/media/
barcelo.com
marriot.com