

Predicting Airbnb User Booking Destinations

Lisa Oshita
Dr. Glanz

May 26, 2018

About the Competition

- ▶ Recruiting competition hosted on Kaggle from November 2015 to February 2016
- ▶ **Task:** build a model to predict where new Airbnb users will book their first destinations
- ▶ **12 possible destinations to predict:** Australia, Canada, France, Germany, Italy, Netherlands, Portugal, Spain, United Kingdom, US, other and no destination found (NDF)

About the Data

▶ **train_users**

- ▶ 213,415 observations and 16 rows
- ▶ Contains information about users from 2010 to 2014

▶ **sessions**

- ▶ 1,048,575 rows and 12,994 unique users
- ▶ contains information about web session activity for each user
- ▶ 10% of rows from each unique user were randomly sampled
- ▶ New sampled data contained 104,424 rows

Booking Destinations: Extremely Imbalanced Classes

Destination	Percentage of the data (%)
NDF	58.35
US	29.22
other	4.73
FR	2.35
IT	1.33
GB	1.09
ES	1.05
CA	0.67
DE	0.50
NL	0.36
AU	0.25
PT	0.10

Table: Percentage of data each destination accounts for

Models

- ▶ Extreme Gradient Boosting (XGBoost)
- ▶ Random forest
- ▶ Stacked model

Feature Engineering

- ▶ Date features
- ▶ Age and gender
- ▶ Count features created from the sessions data (e.g. number of times a user viewed recent reservations, number of times a user viewed similar listings...)
- ▶ Summary statistics of seconds elapsed for each user's web session

Model Building: Feature Importance

- ▶ Full data was split into training (70% of the data) and test sets
- ▶ 5-fold cross-validation was performed with XGBoost and random forest models, with just the top 200 most important features
- ▶ **Accuracy:** 87%
- ▶ Both models only made predictions for US and NDF

Model Building: Feature Importance

XGBoost	Random Forest
firstbook_y.1 age_clean signup_flow firstbook_snspring affiliate_channelother gender_cleanFEMALE gender_cleanMALE age_bucket.1 lag_acb_bin.1..365. firstbook_y2014	firstbook_sn.1 firstbook_wkd.1 firstbook_y.1 lag_acb_binNA lag_acb_bin.1..365. firstbook m.1 lag_bts_bin.1..1369. lag_bts_binNA firstbook_snspring firstbook y2013

Table: Top 10 most import features for each model

Model Building: Accounting for Imbalanced Classes

- ▶ Two techniques:
 - ▶ Oversampling with replacement from under-represented destinations, under-sampling from over-represented destinations
 - ▶ Synthetic Minority Oversampling Techniques (SMOTE) combined with under-sampling from over-represented destinations
- ▶ Under-represented destinations made up at least 4% of the training data after oversampling

Model Building: Results of Oversampling Techniques

► Accuracy: 87%

	AU	CA	DE	ES	FR	GB	IT	NDF	NL	other	PT	US
AU	0	0	0	0	1	0	0	0	1	5	0	22
CA	1	2	0	0	3	0	1	0	1	2	0	14
DE	0	0	0	0	1	1	1	0	0	4	0	14
ES	0	0	1	0	1	1	1	0	1	3	0	17
FR	0	0	2	1	2	0	0	0	0	1	0	9
GB	0	0	0	0	1	2	0	0	0	1	0	8
IT	0	0	1	0	0	0	0	0	1	3	0	7
NDF	0	0	0	0	0	0	0	26154	0	0	0	0
NL	0	0	1	0	5	0	2	0	0	5	0	10
other	1	0	3	1	5	1	1	0	0	6	1	25
PT	0	4	1	0	4	0	3	0	1	7	1	43
US	111	294	213	470	1032	483	586	0	155	2082	43	12930

Figure: Confusion matrix of predictions made with XGBoost on the regular over-sampled training set

Model Building: Stacked Modeling

- ▶ Training data was partitioned into five folds, each containing 20% of the data
- ▶ Build each model on four of the training folds, test on the held-out fold
- ▶ Repeated the process until each fold was used as a test fold
- ▶ Stored those predictions in two columns in the training set
- ▶ Fit each model to the full training set, predict on the test data, store predictions in the test set
- ▶ Final XGBoost fit to the predictions stored in the training set and tested on predictions stored in the test set

Model Building: Results of Stacking

► Accuracy: 87%

	AU	CA	DE	ES	FR	GB	IT	NDF	NL	other	PT	US
AU	0	0	0	3	0	2	2	0	0	11	0	42
CA	0	2	0	2	3	1	2	0	0	7	0	47
DE	0	3	2	2	2	2	2	0	1	6	0	52
ES	0	2	1	4	2	3	1	0	1	7	0	30
FR	0	0	0	0	1	0	0	0	0	0	0	2
GB	0	1	4	1	2	3	0	0	0	5	0	26
IT	0	1	0	3	3	1	4	0	1	1	0	20
NDF	0	0	0	0	0	0	0	37362	0	0	0	0
NL	1	2	1	3	5	0	6	0	0	5	1	35
other	1	2	3	1	8	2	6	0	0	26	0	95
PT	1	2	3	4	7	2	3	0	1	13	0	73
US	158	413	304	651	1473	681	824	0	224	2947	64	18290

Figure: Confusion matrix of stacked model predictions, trained on the regular over-sampled training set

Model Building: Last Steps

- ▶ Perform five-fold cross-validation with XGBoost, random forest, and stacked model on the full data
- ▶ **Accuracy:** 87%
- ▶ Run times:
 - ▶ Random forest: 27.5 min
 - ▶ XGBoost: 28.7 min
 - ▶ Stacked model: 3.24 hr

Discussion: Model Performance

- ▶ Extremely imbalanced classes
- ▶ Base classifiers of the stacked model performed poorly in similar ways

Discussion: Next Steps

- ▶ Stack more than two models
- ▶ Build additional models to predict missing values for certain variables
- ▶ Incorporate the other available data sets
- ▶ Parameter tuning

Thank you!