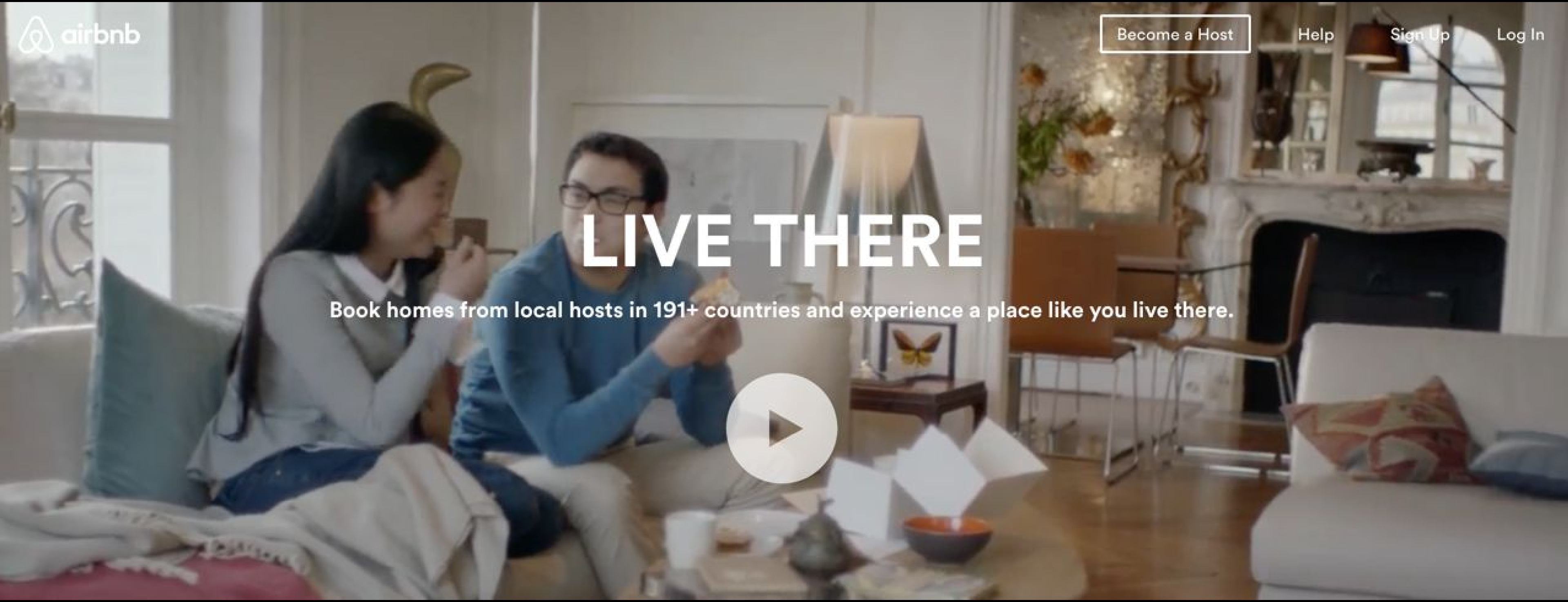


Presented by: Pranati Sahu & Subu Govindaswamy kaggle

Host | Competitions | Datasets | Scripts | Jobs | Community

airbnb

Airbnb New User Bookings

Wed 25 Nov 2015 – Thu 11 Feb 2016 (3 months ago)

Airbnb challenges to predict in which country a new user will make his or her first booking.

Data Sources:

All “United States” user data from 2010 - 2014.

★ train_users.csv

★ sessions.csv

★ countries.csv

★ Age_gender_bkts.csv

Source: https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings

Data Samples

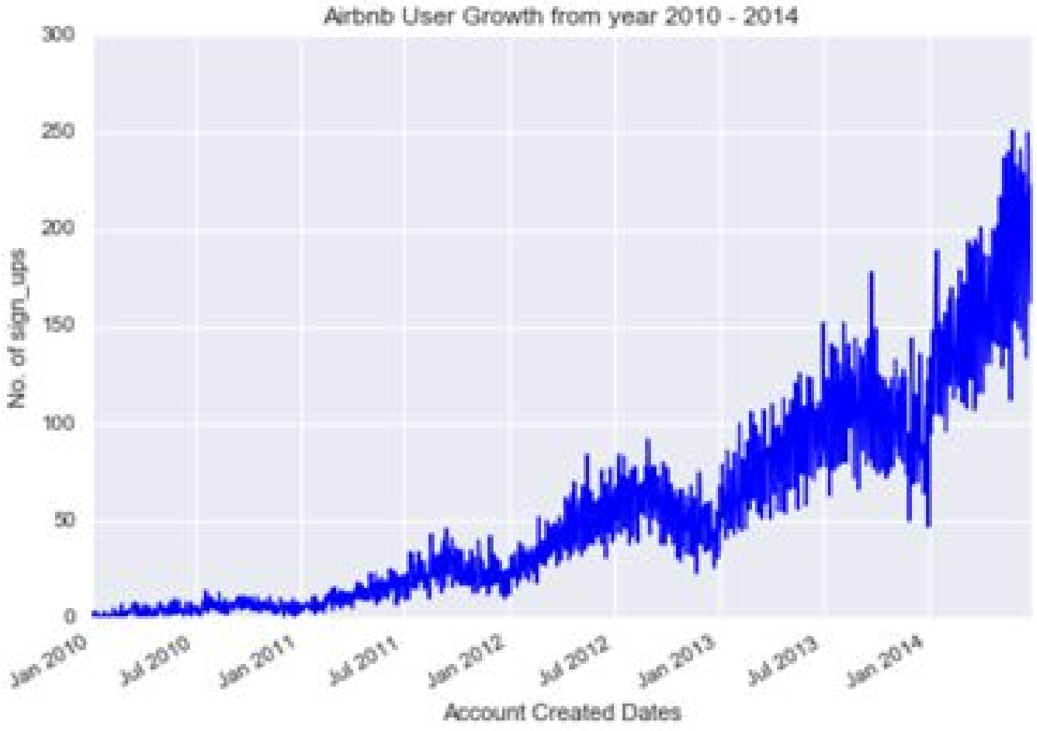
Ugly Slide Alert!!!

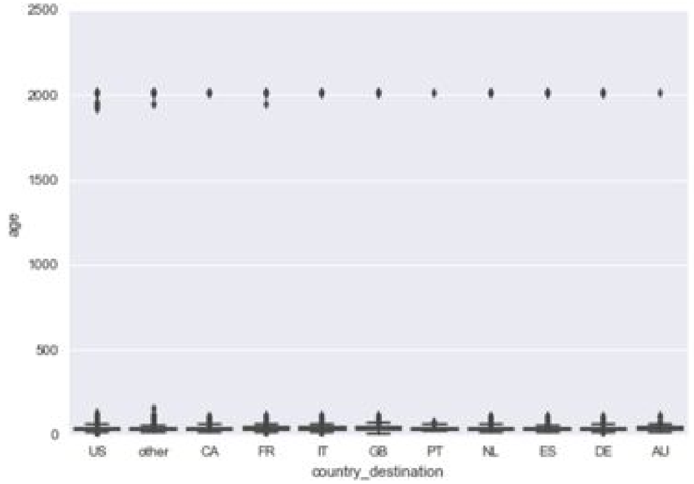
	id	date_account_created	timestamp_first_active	date_first_booking	gender	age	signup_method	signup_flow	language	affiliat
0	gxn3p5htnn	2010-06-28	20090319043255	NaN	-unknown-	NaN	facebook	0	en	direct
1	820tgsjxq7	2011-05-25	20090523174809	NaN	MALE	38.0	facebook	0	en	seo

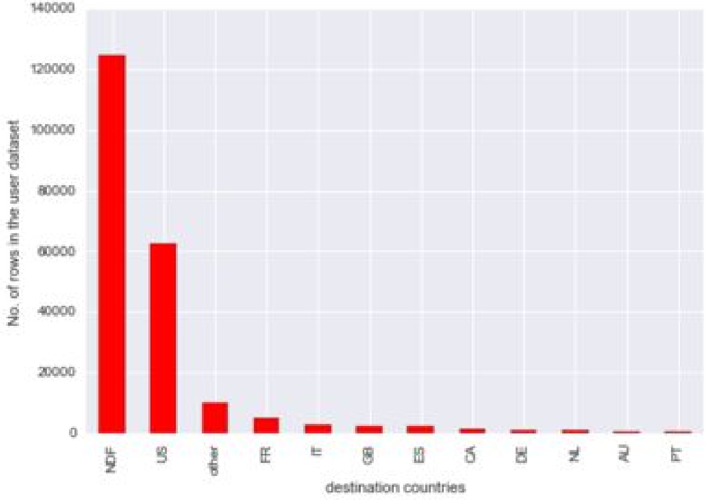
	country_destination	lat_destination	lng_destination	distance_km	destination_km2	destination_language	language_levenshtein_distance
0	AU	-26.853388	133.275160	15297.7440	7741220.0	eng	0.00
1	CA	62.393303	-96.818146	2828.1333	9984670.0	eng	0.00

	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
2	d1mm9tcy42	lookup	NaN	NaN	Windows Desktop	301.0

Exploration Data Analysis

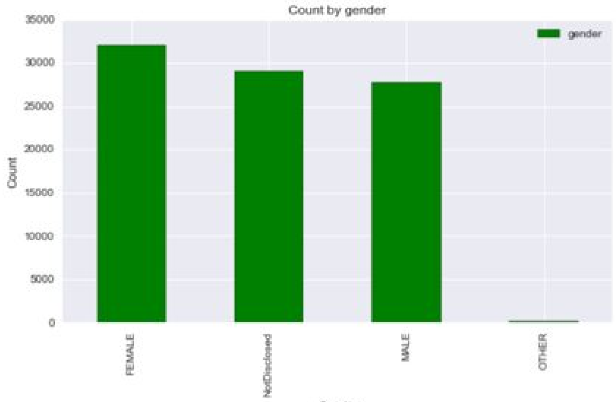


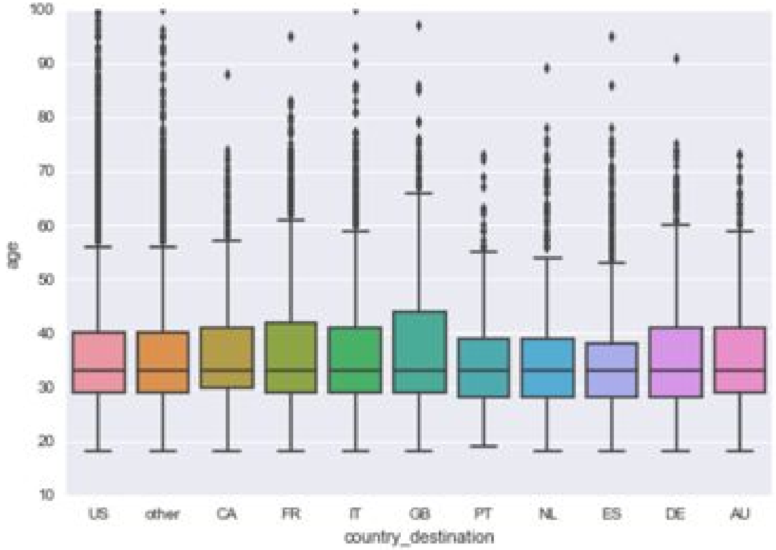




Feature Engineering

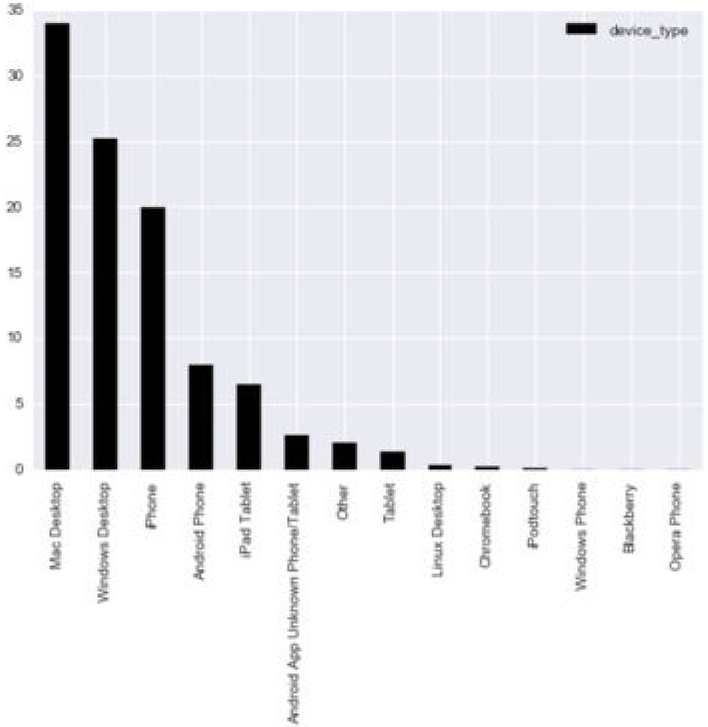
★ ~32% users did not disclose their gender
★ ~30% users did not declare their age

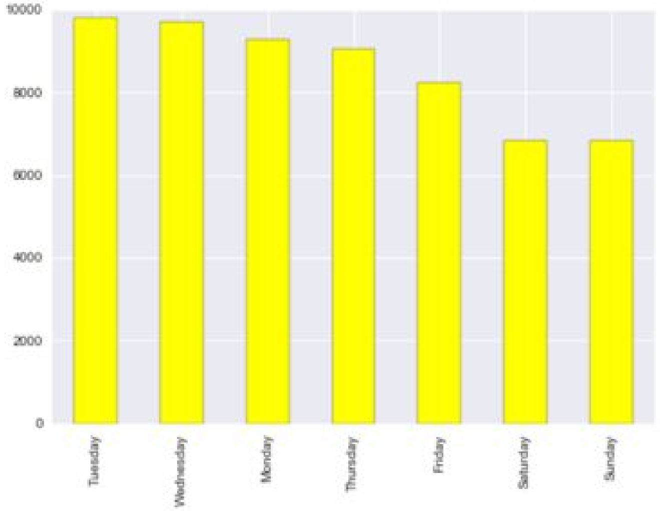


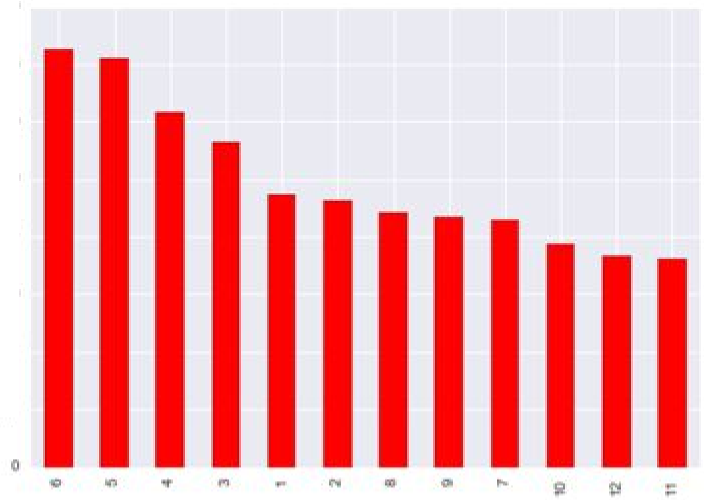


In [45]:
```
# Make a new year/month/day column for date first booking
users['dfb_year'] = users['date_first_booking'].apply(lambda x:x.year)
users['dfb_month'] = users['date_first_booking'].apply(lambda x:x.month)
users['dfb_day'] = users['date_first_booking'].apply(lambda x:x.day)
users['dfb_wn'] = users['date_first_booking'].apply(lambda x:x.isocalendar()[1])
users['dfb_wd'] = users['date_first_booking'].apply(lambda x:calendar.day_name[x.weekday()])
```

Fun Facts


35
30
25
20
15
10
5
0
device_type
Mac Desktop
Windows Desktop
iPhone
Android Phone
iPad Tablet
Android App Unknown Phone/Tablet
Other
Tablet
Linux Desktop
Chromebook
iPodtouch
Windows Phone
Blackberry
Opera Phone


10000
8000
6000
4000
2000
0
Tuesday
Wednesday
Monday
Thursday
Friday
Saturday
Sunday


0
6
5
4
3
1
2
8
9
7
10
12
11

Data Munging

	numcountrdesti	age	signup_flow	Numericgenders	Numersignupmeth	Numuserlangu	Numaffchanne	Numaffprovi	numfirstafftrack	numsignupapp	numfirstdevice	numfirstbrowser
numcountrdesti	1.000000	-0.000744	-0.027955	-0.011209	0.118112	-0.022233	-0.049151	-0.044953	-0.056567	-0.041798	-0.042826	-0.036044
age	-0.000744	1.000000	-0.018821	0.016279	0.060980	-0.009287	-0.000122	-0.000752	0.019883	-0.017294	0.008191	0.019352
signup_flow	-0.027955	-0.018821	1.000000	-0.027329	-0.010516	0.002114	-0.012676	-0.008430	-0.138517	0.750494	0.340099	0.267431
Numericgenders	-0.011209	0.016279	-0.027329	1.000000	0.046583	-0.004360	0.019591	0.012176	0.030261	-0.034624	-0.013382	0.059893
Numersignupmeth	0.118112	0.060980	-0.010516	0.046583	1.000000	-0.054856	-0.106041	-0.078732	-0.027920	-0.066080	-0.029534	0.021911
Numuserlangu	-0.022233	-0.009287	0.002114	-0.004360	-0.054856	1.000000	0.050359	0.043598	0.021415	0.006762	0.018893	-0.003194
Numaffchanne	-0.049151	-0.000122	-0.012676	0.019591	-0.106041	0.050359	1.000000	0.662893	0.264107	0.073464	0.047573	0.008791
Numaffprovid	-0.044953	-0.000752	-0.008430	0.012176	-0.078732	0.043598	0.662893	1.000000	0.252985	0.059056	0.041928	0.042434
numfirstafftrack	-0.056567	0.019883	-0.138517	0.030261	-0.027920	0.021415	0.264107	0.252985	1.000000	-0.130181	0.105574	0.038317
numsignupapp	-0.041798	-0.017294	0.750494	-0.034624	-0.066080	0.006762	0.073464	0.059056	-0.130181	1.000000	0.376181	0.250118
numfirstdevice	-0.042826	0.008191	0.340099	-0.013382	-0.029534	0.018893	0.047573	0.041928	0.105574	0.376181	1.000000	0.632981
numfirstbrowser	-0.036044	0.019352	0.267431	0.059893	0.021911	-0.003194	0.008791	0.042434	0.038317	0.250118	0.632981	1.000000

Correlation Matrix

	numcountrde	age	signup_flow	Numericgende	Numersignupmeth	Numuserlangu	Numaffchanne	Numaffprovid	numfirstafftrack	numsignupapp	numfirstdevic
numcountrdesti	1.000000										
age	-0.000744	1.000000									
signup_flow	-0.027955	-0.018821	1.000000								
Numericgenders	-0.011209	0.016279	-0.027329	1.000000							
Numersignupmeth	0.118112	0.060980	-0.010516	0.046583	1.000000						
Numuserlangu	-0.022233	-0.009287	0.002114	-0.004360	-0.054856	1.000000					
Numaffchanne	-0.049151	-0.000122	-0.012676	0.019591	-0.106041	0.050359	1.000000				
Numaffprovid	-0.044953	-0.000752	-0.008430	0.012176	-0.078732	0.043598	0.662893	1.000000			
numfirstafftrack	-0.056567	0.019883	-0.138517	0.030261	-0.027920	0.021415	0.264107	0.252985	1.000000		
numsignupapp	-0.041798	-0.017294	0.750494	-0.034624	-0.066080	0.006762	0.073464	0.059056	-0.130181	1.000000	
numfirstdevice	-0.042826	0.008191	0.340099	-0.013382	-0.029534	0.018893	0.047573	0.041928	0.105574	0.376181	1.000000
numfirstbrowser	-0.036044	0.019352	0.267431	0.059893	0.021911	-0.003194	0.008791	0.042434	0.038317	0.250118	0.632981

Findings From Correlations

- ★ Highly correlated variables:
  - Signup_App and Signup_Flow (0.75)
  - First_Browser and First_Device
- ★ Moderately correlated variables:
  - First_Browser and Signup_App
  - First_Device and Signup_Flow (0.34)
  - First_Device and Signup_App
- ★ No correlations found
  - Predictors and Response variable (country_destination) (~ 0.00)

Machine Learning Model Used

Logistics Regression

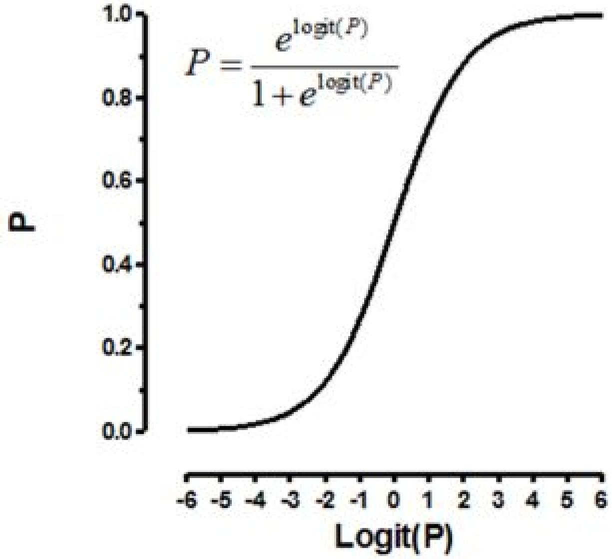


Image source: http://www.analyticsvidhya.com/blog/2015/10/basics-logistic-regression/

Logistic Regression Results

In [184]:
```
logreg = LogisticRegression(C=1e9)
feature_cols = ['age', 'Numericgenders', 'signup_flow', 'Numersignupmeth','Numaffchanne', 'Numuse
rlangu', 'Numaffprovid']
X = ouserdata_NEW[feature_cols]
y = ouserdata_NEW.numcountrdesti
model = logreg.fit(X, y)
model.score(X, y)
```

Out[184]:
```
0.70157916048049673
```

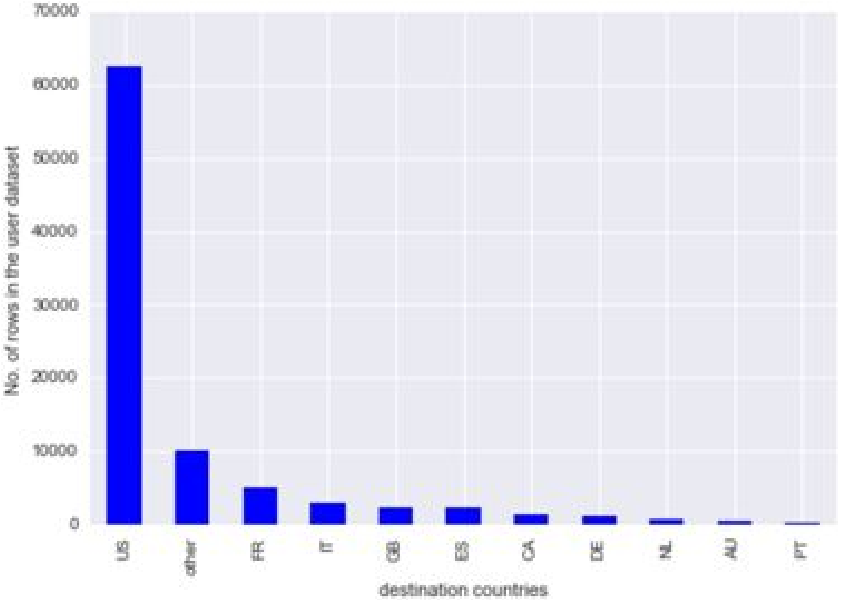
Null Accuracy Rate

The most common country_destination is US.

The null accuracy rate = # of rows with USA divided by total number of rows

= 62376/88908

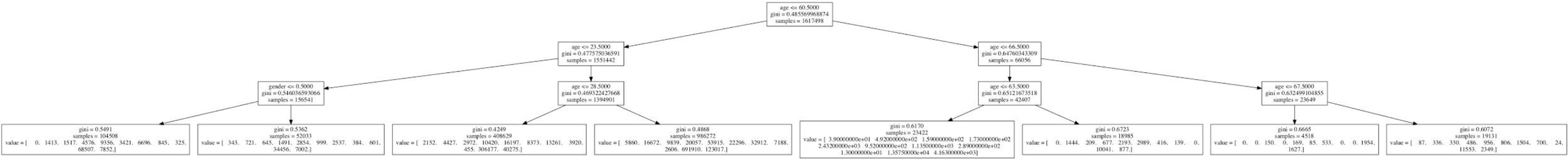
= 70.15%



Machine Learning Models Used

Decision Tree

Random Forest



Decision Tree Results

In [37]:
```
new_df.sort_values(ascending=False, by='importance').head(20)
```

Out[37]:

	feature	importance
0	age	0.143020
461	device_type_Mac Desktop	0.023034
394	signup_app_Web	0.018041
312	dfb_wd_Wednesday	0.017748
341	language_en	0.017284

MAX TREE DEPTH - 10

In [28]:
```
# fit a classification tree with max_depth=10 on all data
from sklearn.tree import DecisionTreeClassifier
treeclf = DecisionTreeClassifier(max_depth=10)
treeclf.fit(X_train, y_train)
```

In [32]:
```
# Calculate Accuracy:
from sklearn import metrics
metrics.accuracy_score(y_test, preds)
```

Out[32]: 0.70405862639542494

Fine Tuned Tree Results

MAX TREE DEPTH - 100

In [6]:
```
# fit a classification tree with max_depth=100 on all data
from sklearn.tree import DecisionTreeClassifier
treeclf10 = DecisionTreeClassifier(max_depth=100)
treeclf10.fit(X_train, y_train)
```

Out[6]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=100,
            max_features=None, max_leaf_nodes=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            random_state=None, splitter='best')
```

In [11]:
```
# Calculate Accuracy:
from sklearn import metrics
metrics.accuracy_score(y_test, preds10)
```

Out[11]:
```
0.97947463318184314
```

	feature	importance
0	age	0.114985
445	secs_elapsed	0.034054
411	first_browser_Chrome	0.023988
391	first_affiliate_tracked_untracked	0.023972
434	first_browser_Safari	0.022352
416	first_browser_Firefox	0.021238
385	first_affiliate_tracked_linked	0.020587
364	affiliate_channel_sem-brand	0.016690
388	first_affiliate_tracked_omg	0.015087
365	affiliate_channel_sem-non-brand	0.014892
314	gender_MALE	0.014197
317	signup_method_basic	0.013681
318	signup_method_facebook	0.013644
315	gender_NotDisclosed	0.013616
313	gender_FEMALE	0.011983
418	first_browser_IE	0.011955
375	affiliate_provider_google	0.009908
402	first_device_type_Windows Desktop	0.009634
366	affiliate_channel_seo	0.009173
371	affiliate_provider_direct	0.008873

Confusion Matrix

Predicted ->		AU	CA	DE	ES	FR	GB	IT	NL	PT	US	other
Actuals:	AU	2937	1	0	3	19	0	2	0	0	83	43
	CA	6	9566	11	19	29	7	22	0	1	174	73
	DE	0	7	5702	0	24	2	7	0	0	130	51
	ES	6	21	0	16111	18	19	27	9	0	376	79
	FR	17	18	19	27	35296	11	149	13	0	770	287
	GB	0	8	2	16	5	15459	36	4	0	356	96
	IT	6	17	6	26	158	28	24441	37	11	692	142
	NL	0	2	0	5	15	4	38	5292	0	196	16
	PT	0	1	0	0	0	13	12	4	1502	34	16
	US	69	188	122	442	734	394	806	194	42	413653	2150
	other	31	53	40	64	290	124	188	39	6	2169	77376

Random Forest Results

```
In [14]: # compute the out-of-bag R-squared score
         rfreg.oob_score_
Out[14]: 0.9311048277627869l
```

```
In [16]: dr15.sort_values(ascending=False, by='importance').head(20)
Out[16]:
```

	feature	importance
0	age	0.190175
445	secs_elapsed	0.069931
411	first_browser_Chrome	0.029841
314	gender_MALE	0.028738
391	first_affiliate_tracked_untracked	0.028162
313	gender_FEMALE	0.027955
385	first_affiliate_tracked_linked	0.026072
315	gender_NotDisclosed	0.025224
416	first_browser_Firefox	0.022426
434	first_browser_Safari	0.020270
388	first_affiliate_tracked_omg	0.018865
317	signup_method_basic	0.015730
318	signup_method_facebook	0.015382
364	affiliate_channel_sem-brand	0.014952
461	device_type_Mac Desktop	0.013279
399	first_device_type_Mac Desktop	0.013251
375	affiliate_provider_google	0.012960
402	first_device_type_Windows Desktop	0.012765
465	device_type_Windows Desktop	0.012628
371	affiliate_provider_direct	0.012212

Model Comparison

- Pros:
  - Decision Tree is very visual and can easily be interpretable.
  - Random Forest was faster than Decision Tree.
- Cons:
  - Decision Tree is very slow. Needs lot of computing power.

Future Scope

1. Lot more feature engineering (Season, Duration, Gender, Age etc.)
2. Removing features that are not important might help with the prediction and model run time
3. Random Forest might provide better result
4. Ensemble might help with accuracy further

For more info :

https://github.com/psahu/GA_Project

https://github.com/subuone/sfdat22_work

Q & A