AIRBNB New User Bookings

Introduction and Problem Statement

Airbnb is an online platform that allows their users to browse through a selection of residences across the world. Now with more than 150 million users on their platform, they have gathered millions of travel booking records from their users. With the data captured, is it possible to predict which country the users will choose based on our dataset? And with this information, what kind of business recommendations and action plans can we provide to help Airbnb succeed in their endeavor of offering customized travel plans, reduced average time for first booking & be better at forecasting demand?

The dataset we used to tackle this problem was obtained from Kaggle and consisted of variables such as demographic information, the date of account creation, date of first booking, country of destination, and how the customer signed up or found out about Airbnb.

Exploratory Analysis

As a starting point, we thought it would be insightful to do some exploratory analysis to see the story that our dataset tells us. Out of 213k data rows with 17 variables; the users in our dataset were from the USA, i.e. the origination point is USA for all travels. We would like to see some predictive pattern emerge from our exploratory analysis to help us with our predictive modelling. Looking at demographic information first, we immediately discovered how disproportional our country of destination was. Looking at Appendix, we see that the dataset is dominated by US and NDF (No destination found).

- The variable 'age' had ages as low as zero to as high a number as 110; for gender the 'Other' category hardly had enough data points. We decided to go ahead with people above the age of 18 (who are legally eligible to book a vacation) and below the age of 70 using some rational logic. We further went ahead and did a box plot comparison between age and gender to derive an insight that irrespective of the gender the median age group for all of them was between 28 to 35.
- Moving on further to analyze the sign-up method we realized that the basic sign up method was leading the chart for most of the countries
- Analyzing the age category further we could see that the age group of 20-24,25-29 & 50-54 have the highest no. of people interested in going for a vacation
- We wanted to see the countries which have the maximum no. of interested people; US clearly was the winner here followed by Denmark & Italy.
- We wanted to see check the seconds elapsed variable which we compared with the gender; Female & Unknown genders seem to be spending most amount of time browsing

- A second check on missing values revealed that only seconds elapsed seem to be high in number and could also be removed as a column since it has 198k missing values out 213k total rows
- We also conducted a t test to compare the average values of the two data sets and determine if they came from the same population and check for unknown variances.
- An ANOVA test was also conducted on signup method and age to test the normality /variance criteria

Feature Selection & Engineering:

- We tested multiple features using chi square to see their impact on the target variable (country destination)
- We used gender, signup method, language, affiliate provider, date of account creation and first device type to check for significance
- We finally decided to zero down on our pre modelling data frame by dropping columns which either had too many missing values or did not provide any rational information ('secs_elapsed', 'first_affiliate_tracked','timestamp_first_active',signup_flow',date_account_created','affiliate_channel', 'affiliate_provider','first_browser')
- Finally, we started grouping categories inside each variable to reduce the no. of categories for better modelling. For example, we had 25 different languages and we grouped them in to five categories based on continents

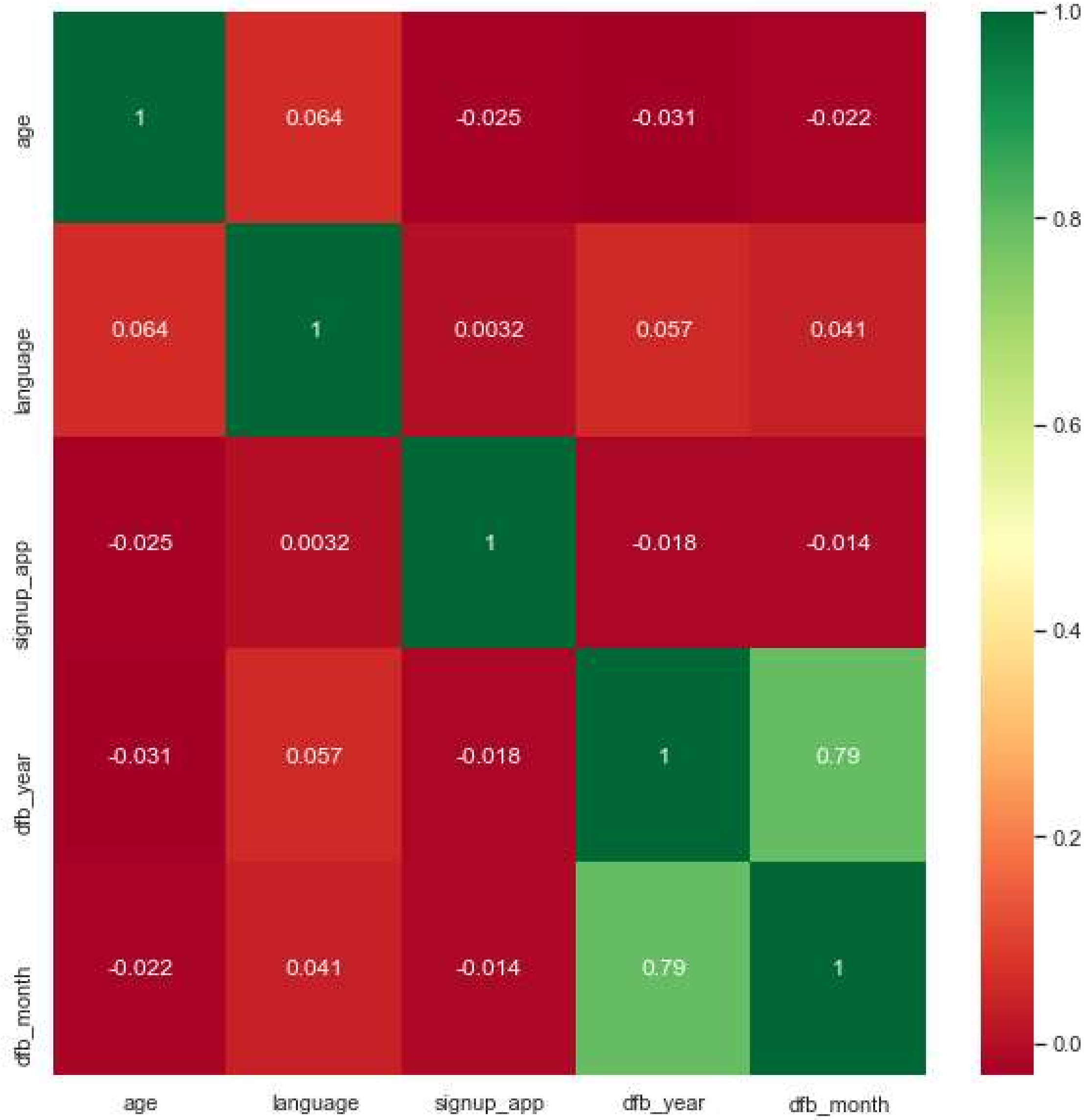
Modelling:

- Since algorithms cannot read textual info, it was time we converted all textual info into numbers, We started with one hot encoding(converting categorical variables in to binomial numerics,i.e. 0/1) for all variables & ran five models using that dataset (Logistic Regression, Support Vector Machine, Decision Tree, Random Forest & Gradient Boost), the results along with accuracy were rather disappointing as the spread of data points were not balanced as far as countries were concerned.
- We then proceeded to tune the hyper parameter for logistic regression and see if it is making any difference; there was no difference post hyper parameter tuning either.
- We resorted to another form of encoding called Label encoding which would assign sequential values to all values within a categorical variable, i.e. If we have three genders, M, F & Others, the encoding will assign values of 1, 2, 3 instead of binary values; We reran the models and this time although the accuracy shot up, all other countries except US & NDF had the negligible data points which made it difficult to believe the accuracy.
- We finally decided to perform a SMOTE (Synthetic Minority Oversampling Technique) where we specified a minimum of 5000 samples for each country which is under sampled. SMOTE-Works by creating synthetic samples from the minor class (no-subscription) instead of creating copies & Randomly chooses one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observations.
- We ran all the models for the third time and saw similar accuracies as the ones post label encoding and slightly better balancing for other countries however this was not enough to accept the model.
- We finally decided that since we are dealing with a multi class problem where in we have 12 different countries to predict from; we should collapse all other countries in to one category as ‘Others’ so that we are left with only three classes, US, Other & NDF where Others has substantial combined data points.

```
In [305]: updated_countries.value_counts()

Out[305]: NDF       54948
          US        47279
          Others    19284
          Name: country_destination, dtype: int64
```

- We ran all the models one last time and finally found justified distribution among the three class variables along with good accuracy; our best two models were Logistic Regression(multinomial) & Gradient Boost however we would like to go ahead with GB based on its standard deviation which is lower.
- We were also curious to find out which are the features that were of utmost importance in the GB model.
- We further analyzed the top five features and checked for correlations among them



-
- We finally decided to check for odds ratio of each predictor variable towards a country and found that we can predict the probabilities of each feature contributing to choosing a destination. For example, signup method is the biggest impactor for NDF whereas language is for the US & Other countries.

Recommendations

The objective for the below recommendations is to assist Airbnb in forecasting demand better, offer customized promotions based on user preference & reduce the average time for the first booking done by a customer.

- Personalized promotions based on the highest probability of going to a destination for a user.
- Customers with high probability of not going to any destination should be offered customized promotions for the country with the second highest probability. Thus, helping them transition from browsing to booking.
- Customers with equal or closer probabilities of going to all countries should be offered generic promotions.

Appendix

Initial overview of the data

```
#Converting in to a dataframe & inital analysis of missing values, we see that date of first booking, age and first af
# tracked have the most missing values
df = pd.DataFrame(data)
len(df) - df.count()
```

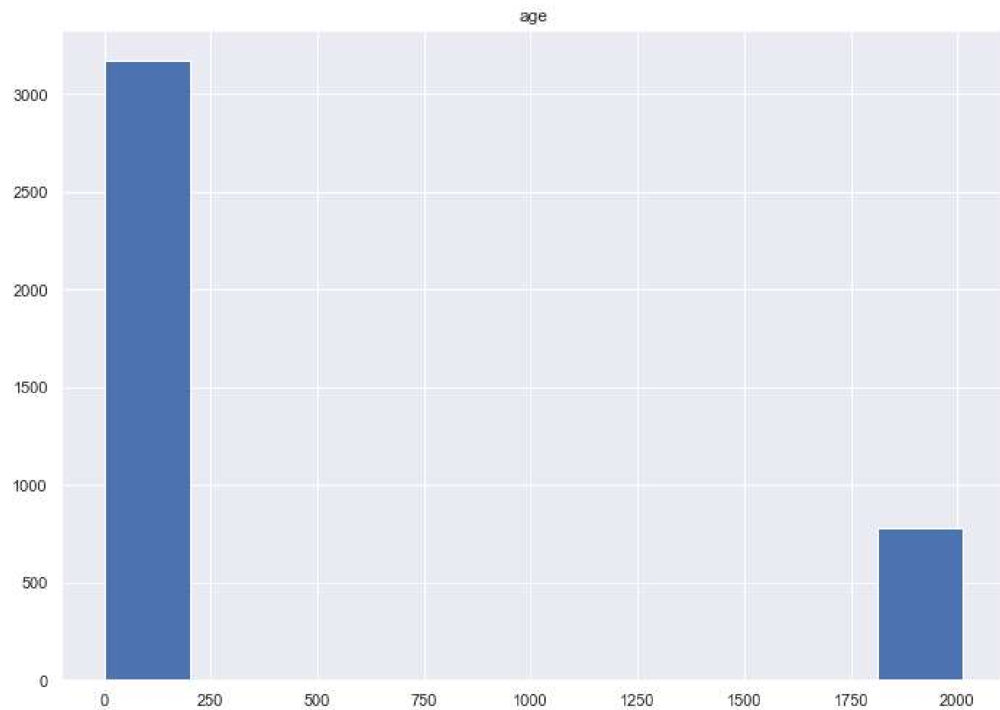
```
id                              0
secs_elapsed               197951
date_account_created            0
timestamp_first_active          0
date_first_booking         124543
gender                          0
age                         87990
signup_method                   0
signup_flow                     1
language                        0
affiliate_channel               0
affiliate_provider              0
first_affiliate_tracked      6065
signup_app                      0
first_device_type               0
first_browser                   0
country_destination             0
dtype: int64
```

```
df['country_destination'].value_counts()
```

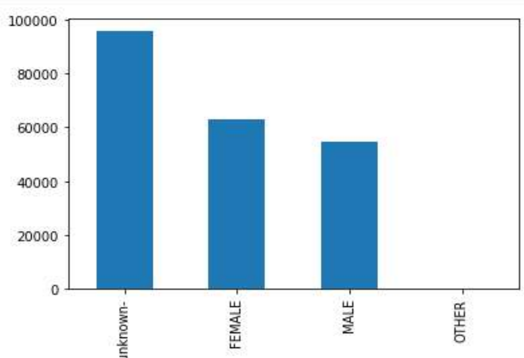
```
NDF      124543
US        62376
other     10094
FR         5023
IT         2835
GB         2324
ES         2249
CA         1428
DE         1061
NL          762
AU          539
PT          217
Name: country_destination, dtype: int64
```

```
#There are people representing ages below 18 who are ineligible to do a booking on Airbnb,
#We also assumed that people above the age of 70 would not be potential tourists
dfcv = data
dfcv = dfcv[(dfcv.age > 70)|(dfcv.age < 18)]
dfcv.hist(column='age')
print(len(dfcv))
```

```
3950
```



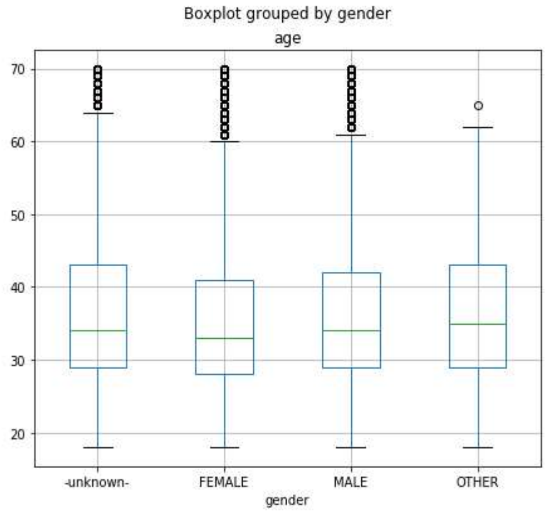

```
#Visualizing the data for all genders
# Unknown gender seems to be leading the charts followed by females and males, Classic case of 'data not missing at rand
dfg = data
dfg = data['gender'].value_counts().plot(kind='bar')
```



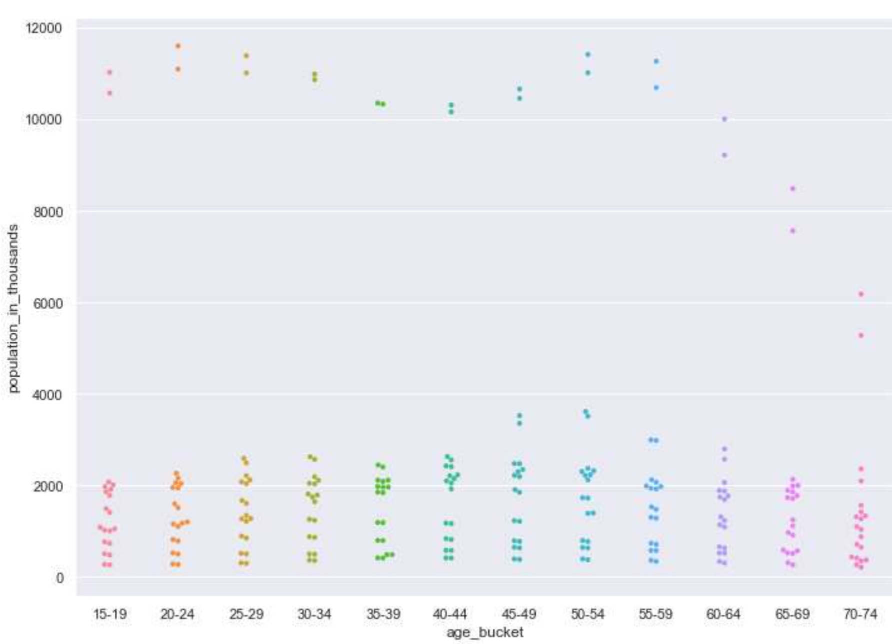
Exploratory Analysis

```
#Comparing Gender versus Age group, we observed that the median age group is around 32 to 35 but are different distributions
#Females, Males & Unknown have pointers which are 1.5 times the upper quartile
#25% of data for Females & Males is greater than the age of 60 & 62 respectively while unknown & Other have an age of
#63 & 65
a = pd.read_csv('C:/Users/laks0/Documents/GitHub/Springboard/Capstone 1/train_users_2.csv', index_col=None)
acd = a[['age', 'gender']]
acd = acd[np.isfinite(acd['age'])]
acd = acd[acd.age >= 18]
acd = acd[acd.age <= 70]
acd.boxplot('age', 'gender', figsize=(7,6))
```

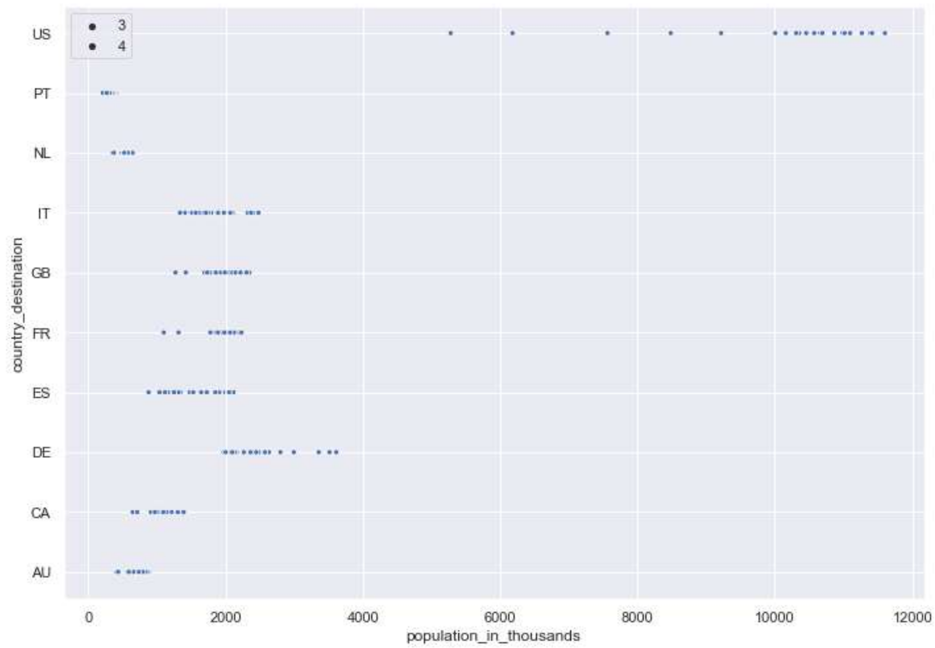
]: <matplotlib.axes._subplots.AxesSubplot at 0x1c8bf4dcb00>



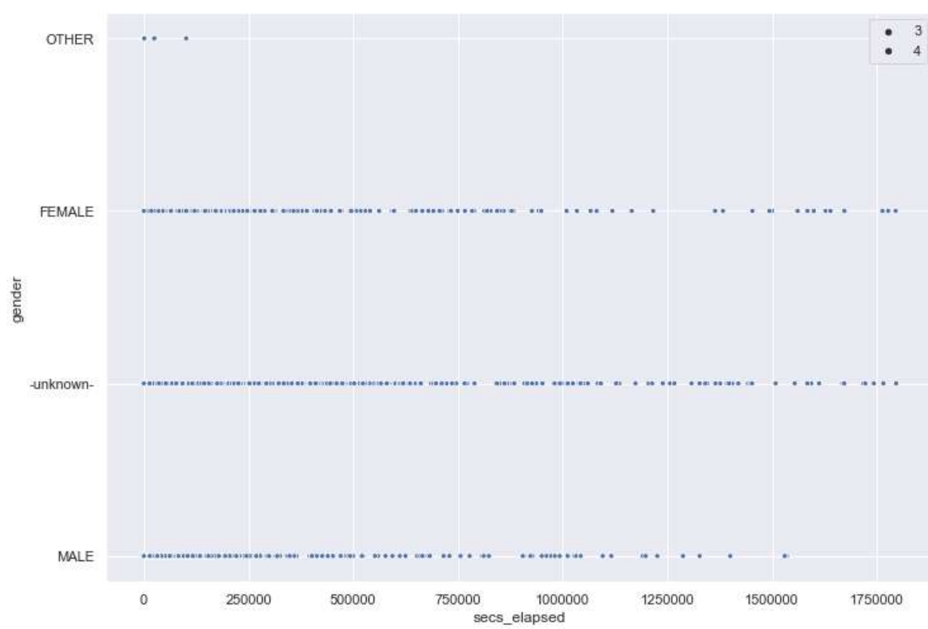

```
#Plotting the no. of people based on age range, we can see that the age group of 20-24,25-29 & 50-54 have the highest no.
# interested in going for a vacation
sns.set(rc={'figure.figsize':(11.7,8.27)})
agebplot = sns.swarmplot(x='age_bucket', y='population_in_thousands', data=age_b, size = 4)
```



```python
#Plotting the countries most people are interested in going to for a vacation; US clearly is the winner here followed by
# Denmark & Italy
sns.set(rc={'figure.figsize':(11.7,8.27)})
countryplot = sns.scatterplot(x='population_in_thousands', y='country_destination', data=age_b, size = 4)
```




```python
#Plotting the gender vs time spent on the website, Female & Unknown genders seem to be spending most amount of time browsing
sns.set(rc={'figure.figsize':(11.7,8.27)})
timeplot = sns.scatterplot(x='secs_elapsed', y='gender', data=data, size = 4)
```



Conducting Chi Square

```python
#Creating a contigency table for Chi Square test:
#H0= We believe there is no relationship between signup method and country of destination (are independent of each other)
#H1 = These two variable are not independent of each other
contingency_table = pd.crosstab(
    df['signup_method'], df['country_destination'])
contingency_table
#Each cell in this table represents a frequency count.
```

country_destination signup_method	AU	CA	DE	ES	FR	GB	IT	NDF	NL	PT	US	other
basic	282	738	507	1038	2387	1127	1300	21293	386	91	30899	4748
facebook	141	309	317	623	1214	578	652	33598	194	62	16315	2572
google	0	1	1	1	3	0	2	57	0	1	65	9

```python
from scipy import stats
stats.chi2_contingency(contingency_table)[:3]
#With a p-value < 0.05 , we reject the null hypothesis. There is a strong relationship between
#'signup method' and the 'country of destination' column, we can see that these two variables are not
#independent of each other.
```

```
(8616.825481629532, 0.0, 22)
```

```python
#Creating a contigency table for Chi Square test:
#H0= We believe there is no relationship between the device used to browse/signup and country of destination
#(are independent of each other)
#H1 = These two variable are not independent of each other
contingency_table = pd.crosstab(
    df['first_device_type'], df['country_destination'])
contingency_table
#Each cell in this table represents a frequency count.
```

country_destination first_device_type	AU	CA	DE	ES	FR	GB	IT	NDF	NL	PT	US	other
Android Phone	0	11	4	9	14	2	11	690	2	0	432	65
Android Tablet	0	6	3	8	18	1	11	305	5	2	223	41
Desktop (Other)	3	17	12	7	17	5	6	317	4	0	304	43
Mac Desktop	224	525	452	853	1966	932	999	23560	305	84	23323	3363
Other/Unknown	8	16	7	28	55	24	17	2667	9	2	930	138
SmartPhone (Other)	0	0	0	0	1	0	1	21	0	0	15	1
Windows Desktop	128	379	253	551	1081	533	631	18638	166	51	15511	2662
iPad	32	48	45	102	244	109	161	3478	56	7	2666	472
iPhone	28	46	49	104	208	99	117	5272	33	8	3875	544

```python
from scipy import stats
stats.chi2_contingency(contingency_table)[:3]
#With a p-value < 0.05 , we  reject the null hypothesis. There is visible relationship between
#'first device type' and the 'country of destination' column, we can see that these two variables are not
#independent of each other.
```

```
(1662.6538804271956, 5.641255919580156e-289, 88)
```

#Creating
#H0= We b
#(are ind
#H1 = The
contingen
df['f
contingen
#Each cel
country_de
first_dev
Andro
Andr
Deskt
Ma
Other
SmartPhor
Windows
from scip
stats.chi
#With a p
#'first d
#independ
(1662.653

Conducting ANOVA:

```
#Creating a contigency table for Chi Square test:
#H0= We believe there is no relationship between the device used to browse/signup and country of destination
#(are independent of each other)
#H1 = These two variable are not independent of each other
contingency_table = pd.crosstab(
    df['first_device_type'], df['country_destination'])
contingency_table
#Each cell in this table represents a frequency count.
```

9]:

country_destination first_device_type	AU	CA	DE	ES	FR	GB	IT	NDF	NL	PT	US	other
Android Phone	0	11	4	9	14	2	11	690	2	0	432	65
Android Tablet	0	6	3	8	18	1	11	305	5	2	223	41
Desktop (Other)	3	17	12	7	17	5	6	317	4	0	304	43
Mac Desktop	224	525	452	853	1966	932	999	23560	305	84	23323	3363
Other/Unknown	8	16	7	28	55	24	17	2667	9	2	930	138
SmartPhone (Other)	0	0	0	0	1	0	1	21	0	0	15	1
Windows Desktop	128	379	253	551	1081	533	631	18638	166	51	15511	2662
iPad	32	48	45	102	244	109	161	3478	56	7	2666	472
iPhone	28	46	49	104	208	99	117	5272	33	8	3875	544

```
from scipy import stats
stats.chi2_contingency(contingency_table)[:3]
#With a p-value < 0.05 , we  reject the null hypothesis. There is visible relationship between
#'first device type' and the 'country of destination' column, we can see that these two variables are not
#independent of each other.
```

0]: (1662.6538804271956, 5.641255919580156e-289, 88)

Combining Categories:

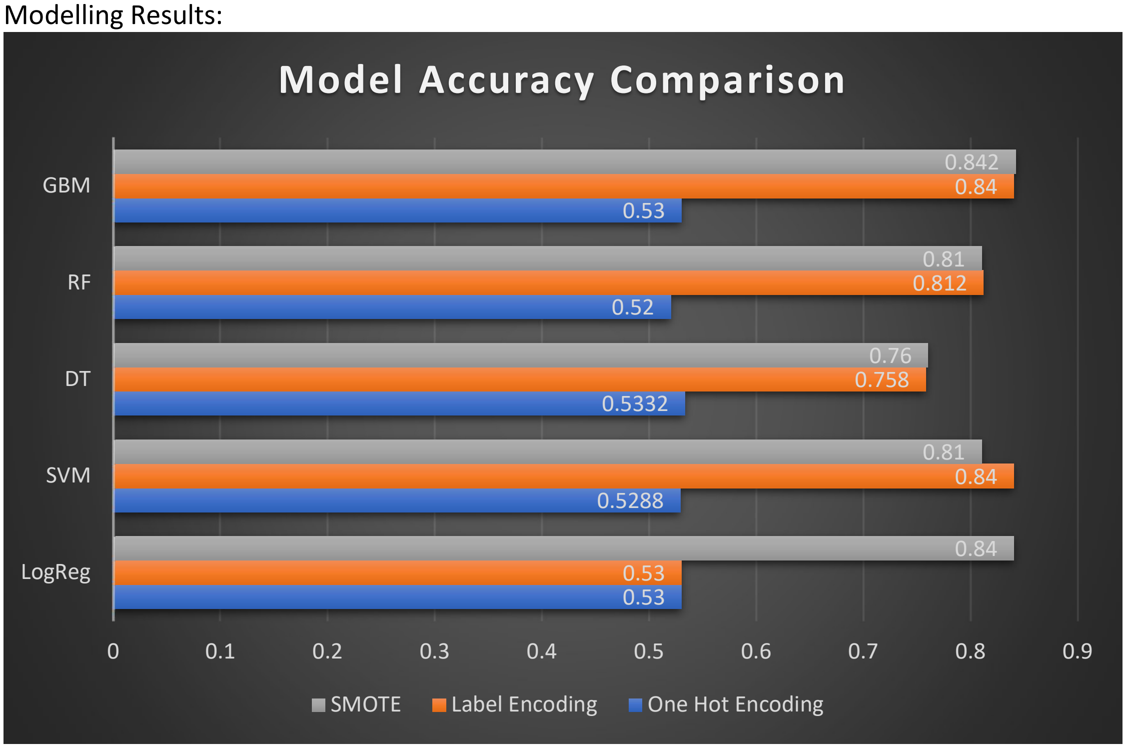
```
#Grouping categories in all the columns to reduce the no. of categories for better modelling:

dffinal['first_device_type']=np.where(np.isin(dffinal['first_device_type'],['Mac Desktop', 'iPhone', 'iPad']),
                                  'Mac', dffinal['first_device_type'])
dffinal['first_device_type']=np.where(np.isin(dffinal['first_device_type'],['Windows Desktop', 'Android Tablet',
                                                                             'Android Phone']),
                                  'Win', dffinal['first_device_type'])
dffinal['first_device_type']=np.where(np.isin(dffinal['first_device_type'],['Other/Unknown', 'Desktop (Other)',
                                                                             'SmartPhone (Other)']),
                                  'Other', dffinal['first_device_type'])
dffinal['language']=np.where(np.isin(dffinal['language'],[ 'zh', 'ko', 'ja', 'id', 'th', 'ca' ]),
                                  'Asian', dffinal['language'])
dffinal['language']=np.where(np.isin(dffinal['language'],[ 'ru', 'sv' ]),
                                  'Russian', dffinal['language'])
dffinal['language']=np.where(np.isin(dffinal['language'],[ 'el', 'hr' ]),
                                  'African', dffinal['language'])
dffinal['language']=np.where(np.isin(dffinal['language'],[ 'de', 'es', 'fr', 'it', 'pt', 'nl', 'pl', 'hu',
                                                         'da', 'fi', 'no', 'tr',
                                                         'cs','is']),
                                  'Eur', dffinal['language'])
dffinal['gender']=np.where(np.isin(dffinal['gender'],['OTHER']),
                                  'unknown', dffinal['gender'])
```

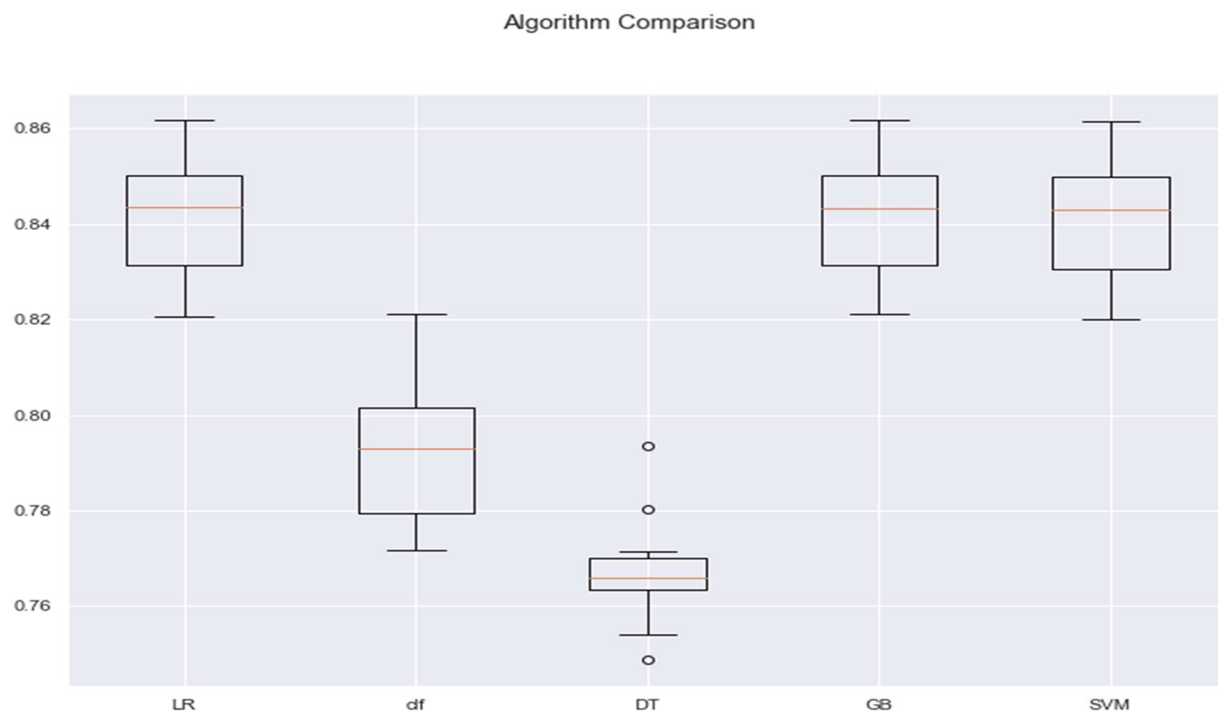
```
#Checking unique values
print(dffinal.language.unique())
print(dffinal.signup_app.unique())
print(dffinal.first_device_type.unique())
print(dffinal.signup_method.unique())
print(dffinal.gender.unique())
```

```
['en' 'Eur' 'Asian' 'Russian' 'African']
['Web' 'Moweb' 'iOS' 'Android']
['Mac' 'Win' 'Other']
['facebook' 'basic' 'google']
['MALE' 'FEMALE' 'unknown']
```

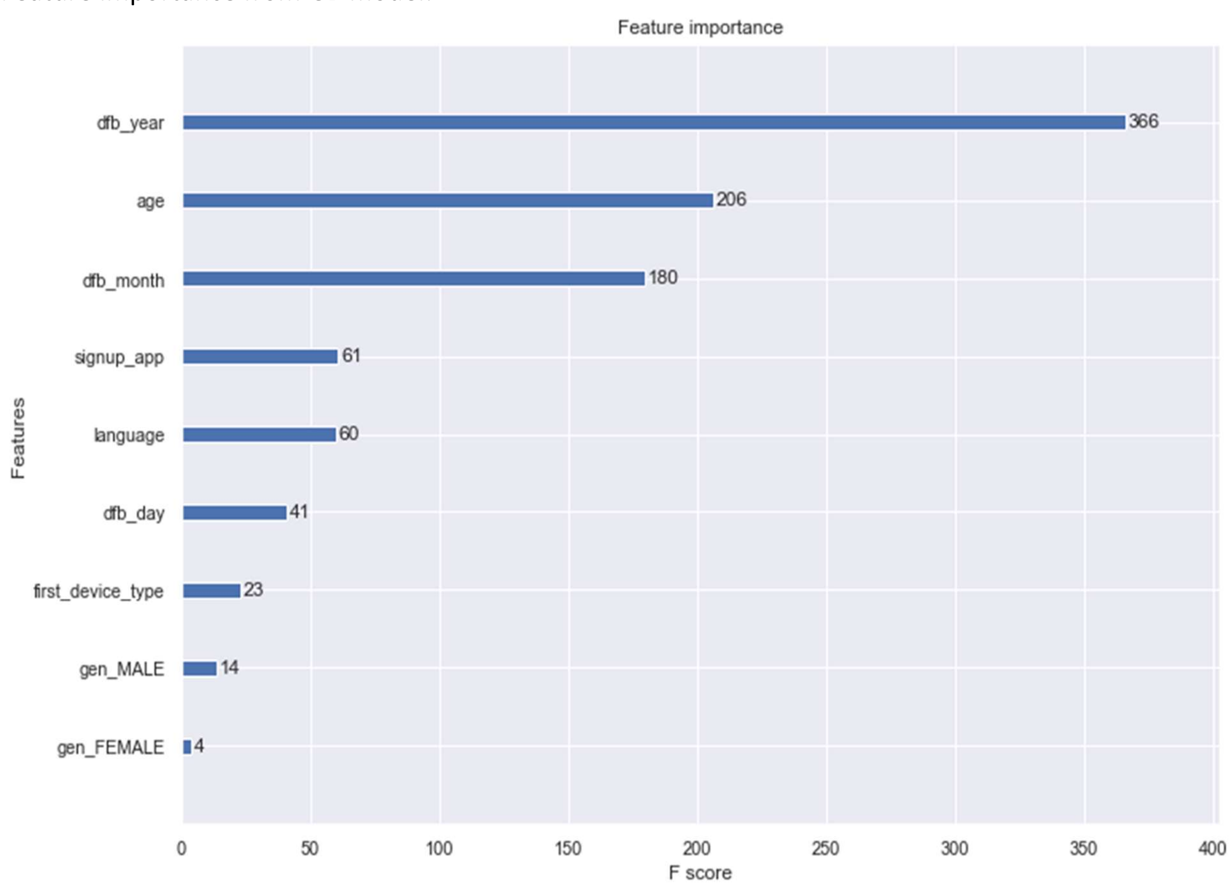
Modelling Results:



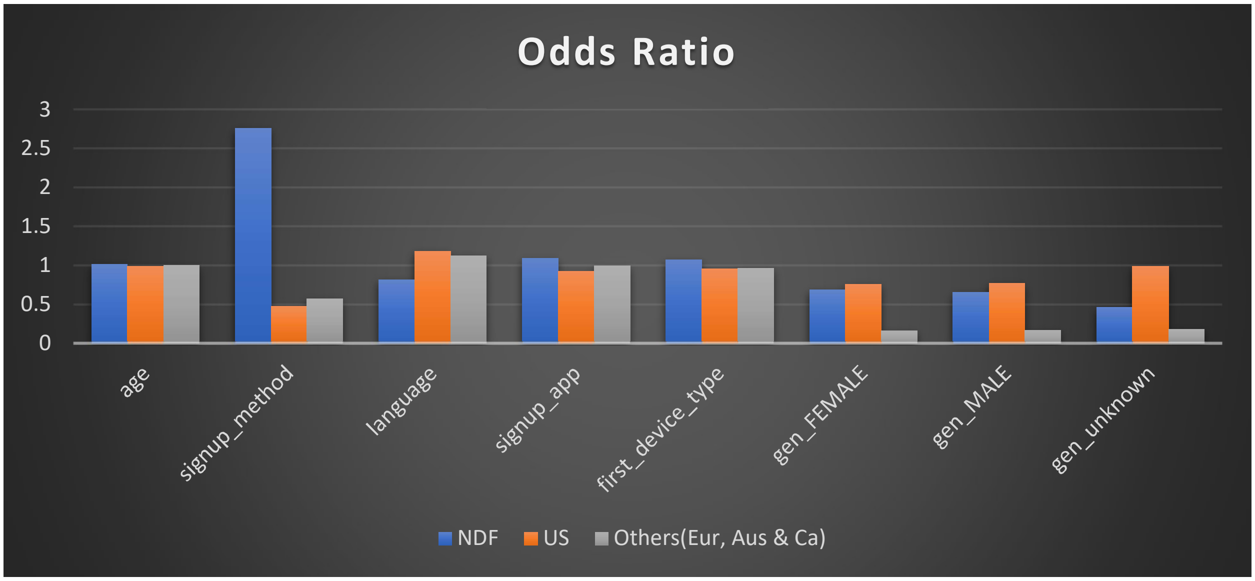
Final Model with only 3 classes of target variable:



Feature importance from GB model:



Odds Ratio:



Features	NDF	US	Others(Eur, Aus & Ca)
age	1.00978	0.990863	0.999007
signup_method	2.756993	0.472126	0.569062
language	0.814498	1.178149	1.120363
signup_app	1.088207	0.924717	0.99234
first_device_type	1.070269	0.952315	0.966298
gen_FEMALE	0.685479	0.753881	0.158632
gen_MALE	0.653397	0.770859	0.166571
gen_unknown	0.460004	0.986122	0.180535