Springboard Data Science Career Track "

Capstone Project #1

Predicting Airbnb First Users' Bookings

By Remzi Kizilboga

June 2020

1. INTRODUCTION

Instead of waking to overlooked "Do not disturb" signs, Airbnb travelers find themselves rising with the birds in a whimsical treehouse, having their morning coffee on the deck of a houseboat, or cooking a shared regional breakfast with their hosts. New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. However, it is not clear where a new user of Airbnb will make a booking activity.

It is clear that by accurately predicting where a new user will book their first travel experience, Airbnb can share more personalized content with their community, decrease the average time to first booking, and better forecast demand. It is also clear that providing more personalized content for Airbnb hosts will help them to make better arrangements which will increase travelers' satisfaction.

This report explains a model developed as a result of the data analysis provided by Airbnb. The data analysis process involves data wrangling, data storytelling, inferential statistics, and modeling processes. A supervised learning model, logistic regression, was applied to classify data. However, even though the training and test accuracy scores were very good, the classification report showed that there was an inbalanced classification. At the end of the data analysis, I created a confusion matrix to explore how the data were classified. The report will end with conclusions and recommendations. The data sets provided by Airbnb can be found at https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings and the data analysis process can be found at https://github.com/remzikboga/springboard_datascience/tree/master/Capstone_Project1.

2. APPROACH

2.1. Data Acquisition and Wrangling

As aforementioned, the data was a part of a Kaggle Competition. All datasets in the project were provided in CSV format. Four of six datasets are relevant to sessions, countries, age-gender, and train datasets. I first cleaned and merged these four datasets to create a final dataset for my analysis. These datasets were the most updated versions which were updated in 2015 for the competition. One of the datasets was a sample for the submissions and the last one was a test dataset so that I did not include both in the cleaning and merging processes.

After all cleaning and merging processes, I created a data frame for the final dataset, which I entitled it as "dfmergefinal." This final dataset consisted of 30 features (columns) and 3340486 observations as seen in Figure 1.

```
In [54]: dfmergefinal.shape
Out[54]: (3340486, 30)
```

Figure 1. The number of features and observations in the final dataset

Among all 30 features, the country destination was the target variable, user ID was the label variable, and the rest of the features were all independent variables. Figure 2 shows the distribution of destination countries.

```
In [59]: dfmergefinal.country_destination.value_counts(dropna=False)
Out[59]: NDF      1833467
         US       1013036
         other     205497
         FR         90282
         IT         59932
         ES         41473
         GB         39540
         CA         19006
         DE         14512
         NL         12830
         AU          7434
         PT          3477
         Name: country_destination, dtype: int64
```

Figure 2. Distribution of destination countries

I did not do too much cleaning at this part except for some cleaning with age data and exploring other data points. For the age data, I noticed that there were some users' ages less than 18 and more than 150. I thought those data points could be selected mistakenly or intentionally wrong. Therefore, instead of dropping those data points, I decided to keep the people who less than 15

and replace the ages more 150 with NaN values to keep potential useful data in other features. Finally, I saved my final dataset as CSV for further analyses.

2.1. Storytelling and Inferential Statistics

For data storytelling, I aimed at answering 13 questions in total using distributions and graphs. In this section, some of the distributions and graphs were reported:

2.1.1. What countries do users mostly want to travel to?

For about 55% of the users, a destination was not found while about 30% of the users travel to the U.S (see Figure 3). Portugal was the least selected destination country among all.

```
NDF      54.886235
US       30.326006
other     6.151710
FR        2.702661
IT        1.794110
ES        1.241526
GB        1.183660
CA        0.568959
DE        0.434428
NL        0.384076
AU        0.222542
PT        0.104087
Name: country_destination, dtype: float64
```

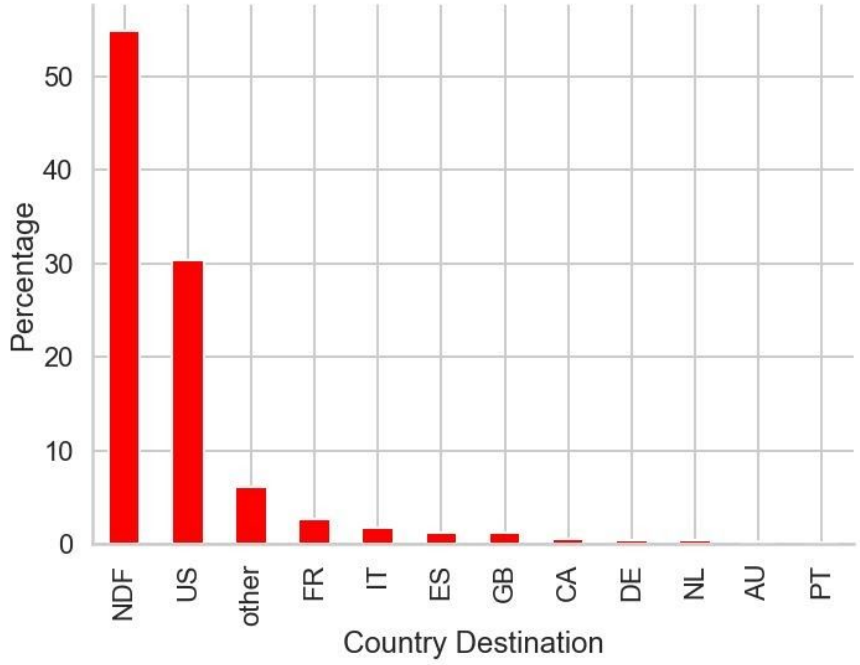

Figure 3. Normalized distribution of user based on destination countries

2.2.2. What is the distribution of genders who are being active to book a stay?

As seen in Figure 4, about 47% of data was coded as NaN. In addition, while 29% of the users were female, 22% of them were mal**e.**

```
NaN       47.222590
FEMALE    29.443530
MALE      23.225363
OTHER      0.108517
Name: gender, dtype: float64
```

Figure 4. The distribution of users' genders

2.2.3. What is the distribution of the age of the users?

I created a distribution plot (Figure 5) to see the distribution of the users' ages. Most of the users were between 20-40 years old. There were only a few users in the age of 80-100. Also, there were some users over 100 age but I believe those users might not want to enter their original ages. Furthermore, there were some values lower than 18 and I did not think Airbnb would allow them to book on their website.

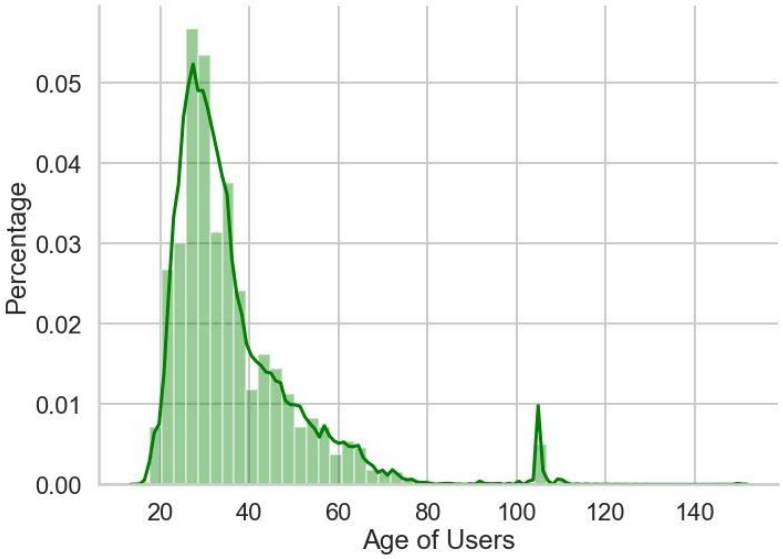

Figure 5. Distribution of users' ages

2.2.4. What is the distribution of the language of the destination countries?

Users mainly preferred English as the language in their booking (see Figure 6).

```
en    96.405673
zh     0.984468
ko     0.542855
fr     0.495796
es     0.429668
de     0.230206
it     0.205000
ru     0.190122
pt     0.176711
ja     0.121809
sv     0.046939
nl     0.036791
da     0.031612
```

Figure 6. Distribution of preferred language

2.2.5. What device do users mostly use for booking?

As seen in Figure 7, about 42% of the users did their search and/or booking on a Mac Desktop while 30% of them using a Windows Desktop computer. In terms of smartphones, users mostly used an iPhone (~11%).

```
Mac Desktop           41.706386
Windows Desktop       30.372796
iPhone                10.797770
Other/Unknown          7.556445
iPad                   7.068642
Android Phone          1.136691
Android Tablet         0.734594
Desktop (Other)        0.606109
SmartPhone (Other)     0.020566
Name: first_device_type, dtype: float64
```

Figure 7. Distribution of devices used for booking

2.2.6. What methods do users mostly use to signup for booking a stay?

There were three signup methods in total: 1) basic, 2) Facebook, and 3) Google. The majority of users signup through the basic signup method while 22% of users prefer Facebook.

2.2.7. What browser do users mostly prefer using for booking their stay?

Figure 8 shows the distribution of the browser used by users to book their stay. 74% of the users used one of Chrome, Safari, Firefox, and Mobile Safari as the browser.

```
Chrome              30.632339
Safari              22.330104
Firefox             12.678634
Mobile Safari        8.398958
IE                   7.307320
Chrome Mobile        0.614611
Android Browser      0.251101
Silk                 0.089179
```

Figure 8. Distribution of browser preferred by users for booking

2.2.8. What are the most common actions while booking?

Figure 9 shows the top ten actions done by users while booking their stay. 21.5% of users apply "show" action. I can see that users did "show" action after doing other actions. Therefore, there can be said that "show" is the result of any other actions.

```
show                    21.513067
search_results           9.155315
personalize              9.108016
index                    7.955459
ajax_refresh_subtotal    6.272261
similar_listings         5.726951
lookup                   3.763644
update                   3.721524
search                   3.275392
social_connections       2.382288
Name: action, dtype: float64
```

Figure 9. Distribution of the most common actions happened during bookings

2.2.9. What days of a week do users mostly become active for booking a stay?

As seen in the bar plot below, users are mostly more active on Wednesdays and Thursdays and less active on weekends. Then, we can say that users will more probably book on weekdays than weekends.

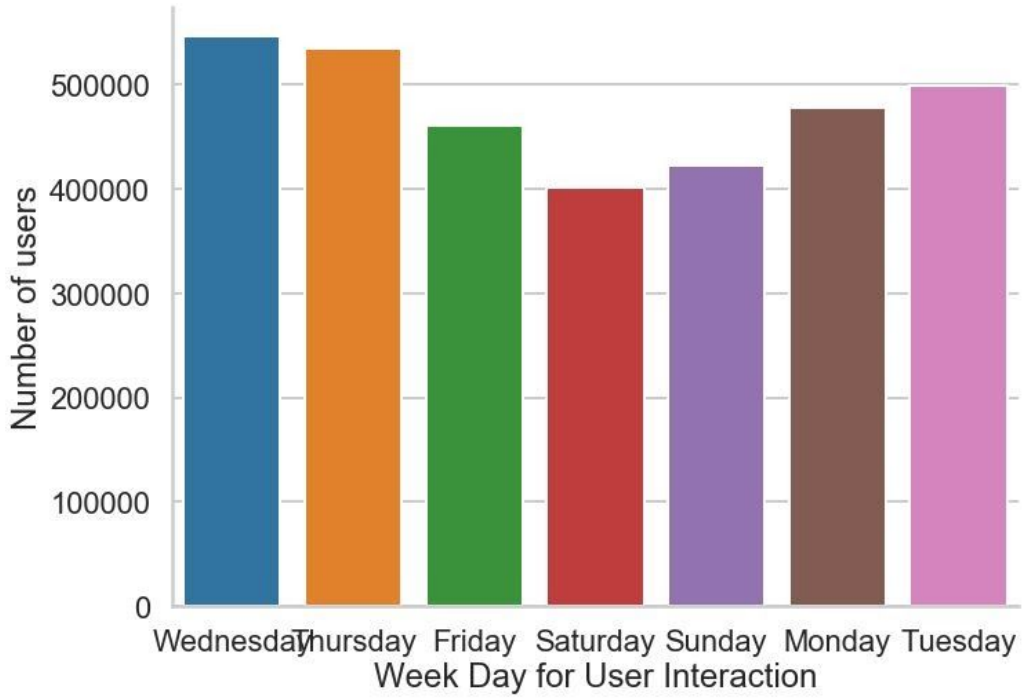

Figure 10. Distribution of weekdays for user bookings

2.2.10. What are the most common applications that users mostly use to signup based on destination countries?

Among non-defined countries and the US, Web is the mostly-used signup method while the Moweb is the least one. As you might remember, we had some other apps as well, which are not even listed here because they don't make too much impact. I did one more thing with the data to check other countries except for No Destination Found (NDF) and I found the same results. So, Web was the most preferred signup app for booking a stay.

2.2.11. What is the most common signup method that users mostly use to signup based on destination countries?

I created a count plot to check the most common signup method preferred by users based on the destination countries where they wanted to book their stay. As seen in Figure 11, among all countries, the basic signup method was the most preferred method while Google was the least one dropping the NDF values.

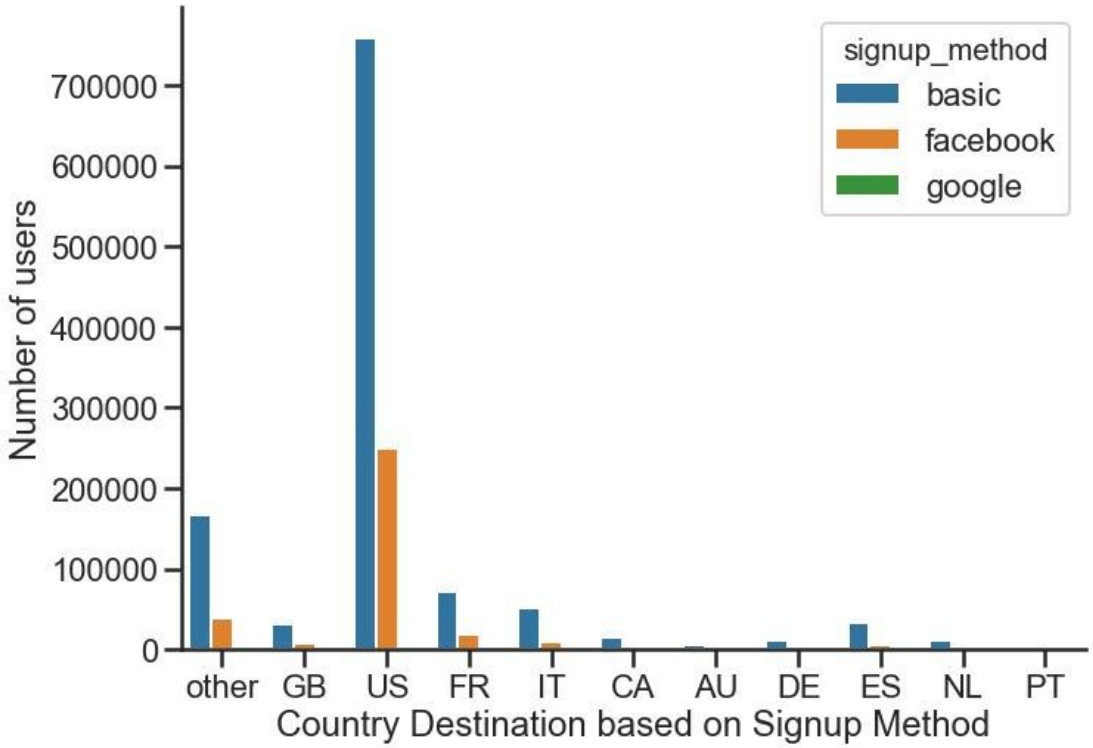

Figure 11. Preferred signup methods based on destination countries

2.2. 12. What is the relationship between users' ages and destination countries?

The boxplot in Figure 12 shows the relationship between users' ages and the destination countries that users wanted to book their stay in. The medians are very close to each other for all destination countries. Users who wanted to book a stay in the US were the youngest ones compare to other countries. Also, users who planned their stays in Great Britain, France, Italy, and Netherland tended to be older. Users for Canada and Australia also tended to be younger.

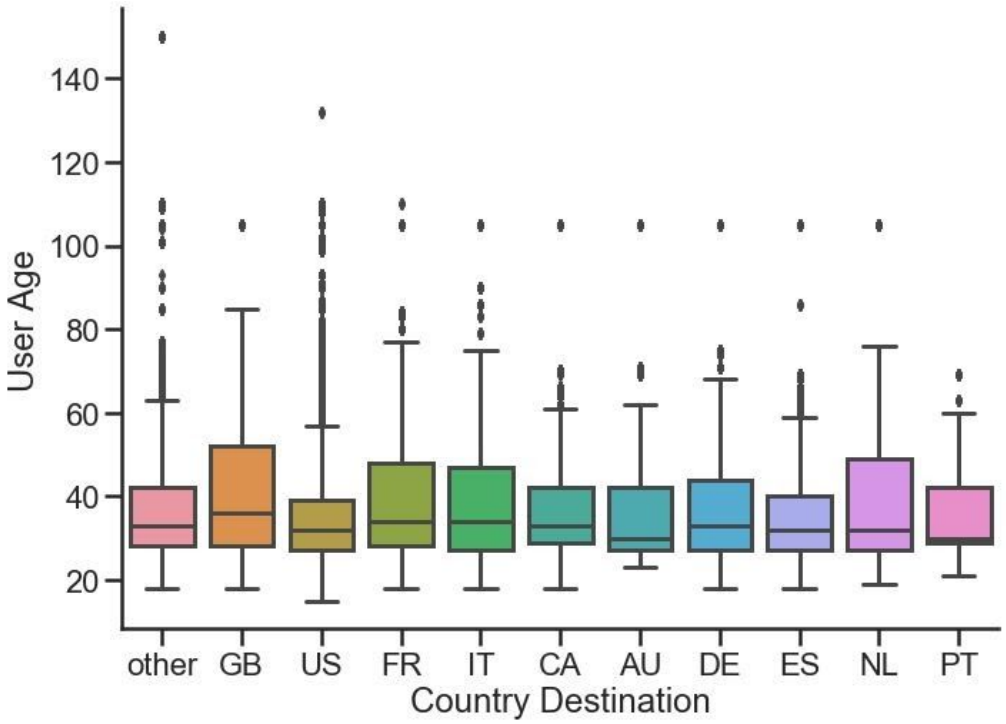

Figure 12. Relationship between users' ages and country destination

2.2.13. What days of a week that users mostly prefer booking based on destination countries?

I created two separate count plots to see what days of a week that users preferred booking their stay based on the destination countries they preferred. Users who wanted to book a stay in the US were usually active on Wednesdays and Thursdays while less active on weekends. Then, I wanted to look at the results one more time without the US because it seemed to me that there could be some interesting results for other destination countries. Therefore, I created another count plot without the US and found that there were some interesting results. When users selected France as their destination country, they became less active on Monday and Friday while more active Wednesday and Sunday. They were less active on Friday for Great Britain as well. Users tried to book a stay mostly on Monday for Italy and Spain. For the countries like Netherland and Portugal, they preferred Saturday.

2.2.14. Inferential Statistics

In terms of inferential statistics, I applied some hypothesis tests using p-values, confidence

intervals, and bootstrapping methods. First, I checked whether or not there was a significant difference between the users' ages in terms of their ages specifically in the US. Users booked in the US are mostly female. The null hypothesis was "there is no difference between males and females who select their country destination as the US in terms of their ages". The alternative hypothesis was "there is a difference between two groups." The t-test results showed that because the p-value was less than 0.05, I rejected the null hypothesis and obtain the alternative hypothesis which showed that there was a significant difference between the means of males and females ages. I also found that

```
In [19]: #calculate t value manually
         n0 = len(male_age)
         n1= len(female_age)
         std0 = male_age.std()
         std1= female_age.std()
         mean0 = mean_male
         mean1= mean_female
         sp = np.sqrt( ((n0-1)*(std0)**2 + (n1-1)*(std1)**2)/ (n0+n1-2) )
         t_ = (mean1 - mean0)/(sp * np.sqrt(1/n0 + 1/n1))
         print(t_)

         -7.118804078367921

In [20]: # Use 0.05 Significance level in two sample t-test
         t_val=((male_age_mean - female_age_mean)-0)/SE
         print(t_val)

         8.056237429570178

In [21]: #calculate p value manually
         p_value = (1 - t(n0 + n1 - 1).cdf(t_)) * 2
         p_value

Out[21]: 1.9999999999989102

In [22]: #calculate t and p values using scipy
         ttest_ind(male_age, female_age)

Out[22]: Ttest_indResult(statistic=7.118804078366678, pvalue=1.0897469575308629e-12)
```

Figure 13. T-test results for relationship between user age and gender

Similarly, I tested whether or not a significant difference between users' genders and the elapsed times that they spent while booking their stay. Similar to the ages-genders relationship, I found out that there was a significant difference between males and females in terms of elapsed time (Ttest_indResult(statistic=4.487927633810977, pvalue=7.1932208754725195e-06)).

Finally, I checked if there was a significant difference between elapsed times in terms of genders using bootstrap and confidence interval. When performed 1000 replicates immediately after setting the random seed to 47, I got the value 22456.2 as the lower limit at 95% confidence interval, which compared almost perfectly the same with the value 22456.6 obtained using the t-distribution confidence interval previously. I also found the 95% confidence interval of the difference in standard deviation between the two groups [3723.4564025594013,

8950.454220856525]. Then, I performed a bootstrapped hypothesis test at the 5% significance level (alpha = 0.05) to calculate the p-value of the observed difference between females and males. I found the p-value as 0.0. Thus, it was sufficiently unlikely that the null hypothesis was true and thus I rejected the null. There was a difference in elapsed time between females and males.

```
In [51]: # # Shifting the Dataset so that the two groups have equal means

         # First calculating the combined mean
         combined_mean = np.mean(np.concatenate((male_bts, female_bts)))

         # Generate the shifted dataset
         male_shifted = male_bts - np.mean(male_bts) + combined_mean
         female_shifted =  female_bts - np.mean(female_bts) + combined_mean

In [52]: # Draw the bootstrap replicates from the shifted dataset
         bs_replicates_male = draw_bs_reps(male_shifted, np.mean, size=1000)
         bs_replicates_female = draw_bs_reps(female_shifted, np.mean, size=1000)

In [53]: # Get the differences for the bootstrap simulated sample
         bs_differences = bs_replicates_male - bs_replicates_female


         # Get the observed difference from the actual dataset
         obs_diff = np.mean(male_bts) - np.mean(female_bts)
         obs_diff
Out[53]: 1089.7511982369979

In [54]: # Calculate the p-value by comparing the bootstrap replicates against the observed difference of the means
         # The fraction of values WITHIN bootstrap replicates array that meet a certain criteria against the obs_di
         ff

         p = np.sum(bs_differences >= obs_diff)/ len(bs_differences)
         print('p-value =', p)
         p-value = 0.0
```

Figure 14. Calculation of p-values comparing bootstrap replicates against the observed difference of the means

2.3. Baseline Modeling

As the final step with the data, I applied a supervised machine learning model: Logistic Regression. My goal was to train the data and test its performance. Before splitting the dataset into train and test data, I applied a few data cleaning techniques and then created dummy variables for the features. Click to see the notebook. After creating dummies, I defined the first parameter which represented the features and the second parameter to represent the target variable. I came up with **1507019** data points in total while 181 features for the first parameter. The target variable was “country destination” without NDF. Using the train_test_split method of sci-kit learn, I split the 80% of data into the training data, construct the logistic regression model, fit the model on the training data, and calculated the training and test accuracy scores. The

accuracy scores were very good as seen in Figure 15.

```
                                warm_start=False)

In [96]: y_predict_test = clf.predict(Xtestlr)
         print("\n")
         print("[Test] Accuracy score (y_predict_test, ytestlr):",accuracy_score(y_predict_test, ytestlr))

         # Note the order in which the parameters must be passed
         # according to the documentation ... although there should be
         # no difference since it is a one-to-one comparison ...
         # ref: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metric
         s.accuracy_score
         print("\n")
         print("[Test] Accuracy score: (ytestlr, y_predict_test)",accuracy_score(ytestlr, y_predict_test))

         # also printout the training score
         y_predict_training = clf.predict(Xlr)
         print("\n")
         print("[Training] Accuracy score: (ylr, y_predict_training)",accuracy_score(ylr, y_predict_training))


         [Test] Accuracy score (y_predict_test, ytestlr): 0.9564770208756354


         [Test] Accuracy score: (ytestlr, y_predict_test) 0.9564770208756354


         [Training] Accuracy score: (ylr, y_predict_training) 0.9562538621367518
```

Figure 15. Training and test accuracy scores

Even though data accuracy scores showed a perfect model, as exploratory data analysis showed me, there could be an inbalanced classification because the US was the most preferred country destination over other countries. Therefore, I decided to look at classification report to see how the data was classified. My assumption was that the groups which had fewer data points could have lower precision, recall, and F1-scores. However, I had some surprising observations as the result of the classification report as seen in Figure 16. For example, Portugal had perfect scores even though it had the least data classified to. I found out that the data was classified well for Great Britain, Australia, and Canada.

```
In [98]: # use sklearn.metrics.classification_report for a more comprehensive
         # performance analysis


         from sklearn.metrics import classification_report
         # ref: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html#sklear
         n.metrics.classification_report

         print("[Training Classification Report]")
         print(classification_report(ylr, y_predict_training))

         print("[Test Classification Report]")
         print(classification_report(ytestlr, y_predict_test))
```

```
[Training Classification Report]
/Users/remzikboga/anaconda3/lib/python3.7/site-packages/sklearn/metrics/classification.py:1437: UndefinedM
etricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted sampl
es.
  'precision', 'predicted', average, warn_for)
              precision    recall  f1-score   support

          AU       0.54      0.00      0.00      5916
          CA       0.46      0.03      0.06     15257
          DE       1.00      1.00      1.00     11559
          ES       1.00      1.00      1.00     33193
          FR       1.00      1.00      1.00     72218
          GB       0.00      0.00      0.00     31660
          IT       1.00      1.00      1.00     48064
          NL       1.00      1.00      1.00     10216
          PT       1.00      1.00      1.00      2805
          US       0.94      1.00      0.97    810437
       other       1.00      1.00      1.00    164290

    accuracy                           0.96   1205615
   macro avg       0.81      0.73      0.73   1205615
weighted avg       0.92      0.96      0.94   1205615

[Test Classification Report]
              precision    recall  f1-score   support

          AU       1.00      0.00      0.00      1518
          CA       0.49      0.04      0.07      3749
          DE       1.00      1.00      1.00      2953
          ES       1.00      1.00      1.00      8280
          FR       1.00      1.00      1.00     18064
          GB       0.00      0.00      0.00      7880
          IT       1.00      1.00      1.00     11868
          NL       1.00      1.00      1.00      2614
          PT       1.00      1.00      1.00       672
          US       0.94      1.00      0.97    202599
       other       1.00      1.00      1.00     41207

    accuracy                           0.96    301404
   macro avg       0.86      0.73      0.73    301404
weighted avg       0.93      0.96      0.94    301404
```

Figure 16. Classification report code and table

As the final step, I wanted to look at how data was classified because the classification report showed me only the problematic classifications. Thus, I performed a confusion matrix to see the way data was classified. The confusion matrix results showed that most of the data points for Australia (AU), Canada (CA), and all data points for Great Britain (GB) were classified as the US (see Figure 17).

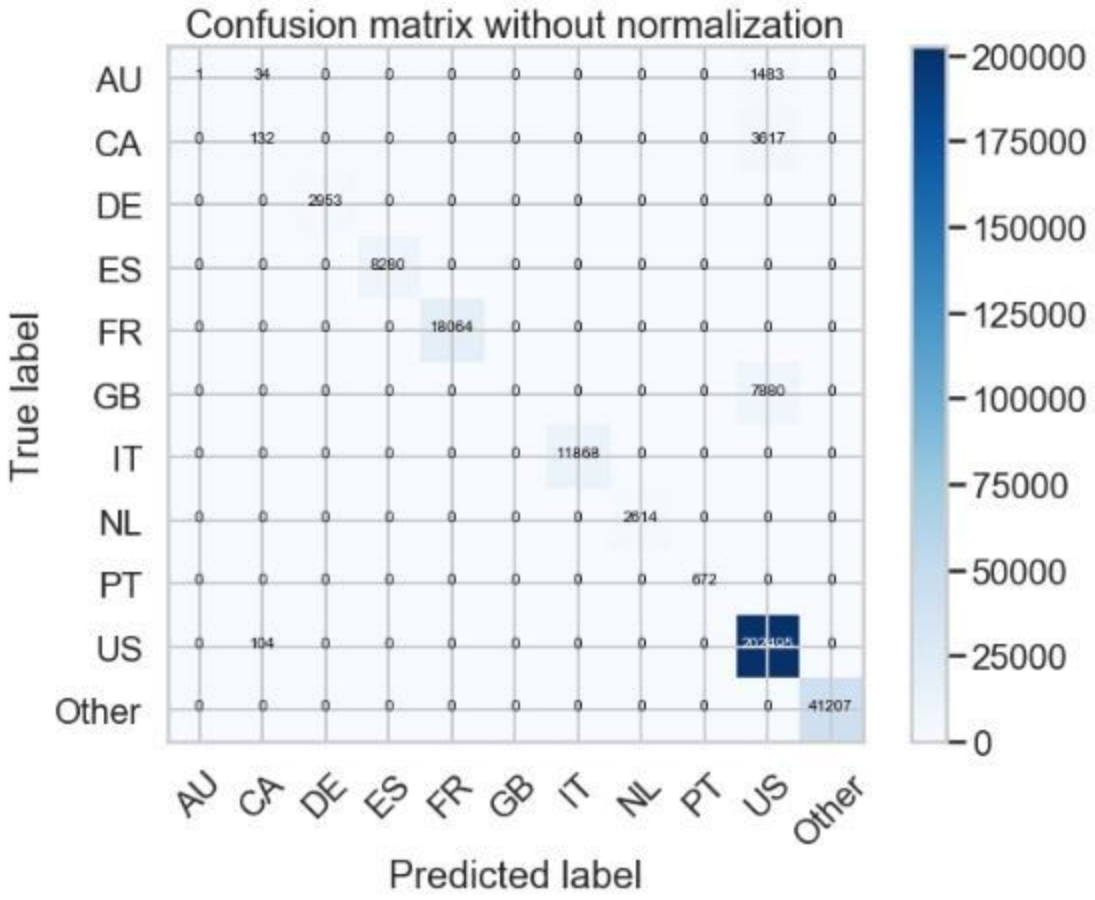

Figure 17. Confusion matrix results

3. Conclusions and Future Work

This report provides an executive summary of the data analysis process of predicting Airbnb's first users' bookings, including data wrangling and cleaning, data storytelling, inferential statistics, and model creation processes. The results showed an inbalanced classification where three countries (GB, AU, CA) were classified as the US. To access the entire analysis process, visit https://github.com/remzikboga/springboard_datascience/tree/master/Capstone_Project1. For future work:

1. The destination countries rather than the US could be treated like "other" countries so that it could be possible to create a model for binary classification.
2. Another classification model can be applied only for the countries excluding the US.
3. The macro average scores were pretty low, and the weighted average scores are much higher. That tells us that the predictions were good on the larger classes and much poorer on the other ones. The model could be improved using another model like XGBoost.

4. Recommendations for the Clients

1. The majority of users use Web over mobile devices for booking. It can be reasonable to say that it is better to invest more in Web applications.
2. The majority of users prefer Apple devices (Macbook, iMac, iPhone, iPad) than other devices. It perfectly makes sense for businesses to invest in improving their systems in Apple devices.
3. In addition, a huge majority of users mostly prefer using the basic signup method over Facebook and Google. The business might want to put more investment in posting ads on Facebook because the social media use for people between 20-40 is very high. https://www.statista.com/statistics/246221/share-of-us-internet-users-who-use-facebook-by-age-group/. This might probably increase the trends in bookings between and among friends.

Resources

- **Itertools: https://docs.python.org/3/library/itertools.html**
- **Matplotlib: https://matplotlib.org/**
- **Pandas: https://pandas.pydata.org/**
- **Pickle: https://docs.python.org/3/library/pickle.html**
- **Seaborn: https://seaborn.pydata.org/**
- **Sci-kit Learn: https://scikit-learn.org/stable/**