Airbnb First Destination Country Classification Project Milestone Report

Project Scope

Project Objective:

The capstone project aims to predict which country will be new users first booking destination so that more relevant content will be displayed on the users interface. The analysis could help to deliver more customized messages to the users across all the touchpoints to increase the booking conversion probability of the new users.

For the analysis, user demographic data, web session records and some summary statistics about the destination countries will be used. The dataset could be downloaded from the link below:

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

Approach for solving the problem:

The whole project will have four main stages:

- Data Cleaning (Clean up all the fields and merge all different data sources into one single analytical dataset)
- Data Exploratory Analysis (Understand the distribution of all variables, identify some of the constraints of the analysis due to the limitation of the data)
- Modeling: This is a classification problem. The methodologies I will try out will include (but limit to) logistics regression, Random Forest, Ada Boost, Gradient Boost and Xg Boost.
- Validation: Leverage the model to predict the testing dataset. The metrics to evaluate the model would be precision, recall, F1 score, ROC AUC and also the accuracy score.
- Presentation: Present insights from the model (eg: important online factors that could impact new users decision on choosing their first destination country)

Deliverables: Scripts for each stage of the project and slides for presenting the methodologies, whole workflow and some additional insights out of the project.

Data cleaning

Raw Data Set:

5 raw datasets:

- sessions.csv - web sessions log for users
    - ○ user_id: to be joined with the column id in users table
    - ○ action
    - ○ action_type
    - ○ action_detail
    - ○ device_type
    - ○ secs_elapsed
- train_users.csv - the training set of users
- test_users.csv - the test set of users
    - ○ id: user id
    - ○ date_account_created: the date of account creation
    - ○ timestamp_first_active: timestamp of the first activity, note that it can be
    - ○ earlier than date_account_created or date_first_booking because a user
    - ○ can search before signing up
    - ○ date_first_booking: date of first booking
    - ○ gender
    - ○ age
    - ○ signup_method
    - ○ signup_flow: the page a user came to signup up from
    - ○ language: international language preference
    - ○ affiliate_channel: what kind of paid marketing
    - ○ affiliate_provider: where the marketing is e.g. google, craigslist, other
    - ○ first_affiliate_tracked: whats the first marketing the user interacted with
    - ○ before the signing up
    - ○ signup_app
    - ○ first_device_type
    - ○ first_browser
    - ○ country_destination: this is the target variable you are to predict
- countries.csv - summary statistics of destination countries in this dataset and their locations
- age_gender_bkts.csv - summary statistics of users age group, gender, country of destination
- sample_submission.csv - correct format for submitting your predictions

Data Cleaning Step:

- Group count of session by user id as count of activity count and sum of “secs elapsed” by user id as total time spent with airbnb website
- Merge user training data with the session data by user_id
- 58% of users have not made a booking yet. So I created the trip_booking_flag field so that before getting into the multi class classification problem to predict the first destination country, I can start with binary classification problem.
- There are some users with unreasonable ages that are lower than 18 or higher than 122. For the users with age that is lower than 18, the age is computed as 18. For the users with age that is higher than 122, the age has been computed as 122.
- Since there are some null values in the age. To resolve the issue, I have cut the age into buckets based on quantiles and null values are categorized into the unknown bucket.
- Since there are some null values in total time spent with airbnb website and total activity count fields, I have cut the fields into bucket based on quantile and the null values are categorized into the unknown bucket.
- For “total time spent” and “activity count” fields, I have also filled the null value with zeros to keep the continuous variables.

Feature Engineering: Creating extra features based on the existing features

- For the account creation date, four more fields have been created: year, month, day and day of week.
- For the first active date, four more fields have been created: year, month, day and day of week.
- For the first active date, four more fields have been created: year, month, day and day of week.
- In the user dataset, there are two timestamp fields: account creation date and first active timestamp. By leveraging these two timestamp fields, two numerical fields have been created: number of days since the account creation as of latest day and number of days since the first activity as of latest day.
- Since there are missing values in the dataset for user age, total time spent (in seconds) and also the session count. The buckets have been created for these fields by cutting the data into quantiles. For the missing values, a bucket called “Unknown” has been created for each of the fields.
- By leveraging the activity count field and action type field from the session dataset, I have created activity counts based on different action types. Some examples of the features include:

Session Count by action name_payment_instruments,
Session Count by action name_payment_methods,
Session Count by action name_payoneer_account_redirect,
Session Count by action name_payoneer_signup_complete,
Session Count by action name_payout_delete,
Session Count by action name_payout_preferences,
Session Count by action name_payout_update,
Session Count by action name_pending,
Session Count by action name_pending_tickets,

Final Cleaned Dataset:

Below are the fields in the final cleaned dataset:

```
Index(['id', 'date_account_created', 'timestamp_first_active_cleaned',
       'gender', 'signup_method', 'signup_flow', 'language',
       'affiliate_channel', 'affiliate_provider', 'first_affiliate_tracked',
       'signup_app', 'first_device_type', 'first_browser',
       'country_destination', 'age_computed',
       'Account_creation_before_booking_flag', 'Total time spent (in seconds)',
       'number_of_active_day_as_of_latest_date',
       'number_of_days_since_account_creation_as_of_latest_date',
       'session count', 'trip_booking_flag', 'Account_creation_date_month',
       'Account_creation_date_year', 'Account_creation_date_day',
       'Account_creation_date_day_of_week', 'first_active_date_month',
       'first_active_date_day', 'first_active_date_year',
       'first_active_date_dayofweek', 'age_bucket',
       'Total time spent (in seconds)_fill_null_zero',
       'session count_fill_null_zero', 'Total time spent (in seconds)_bucket',
       'session count_bucket'],
      dtype='object')
```

Other than fields above, the final dataset also includes features such as activity count by action type.

Exploratory Data Analysis Part One - Variable Descriptive Analysis

User Demographic:

- **What do the users in the dataset look like?**
  - 213,451 users in total from US.
  - 45% of users have unknown gender.
  - 59% of users have unknown age.

User Acquisition Channel:

- **Where do the users come from?**
    - Most users are either coming to the website by typing in URL directly or coming from Google.

User Behaviors:

- **How did the users come to the website for the first time? How long have they been browsing on the website?**
    - Majority of users signed up through desktop.
    - Mac is more popular among users than any other device.
    - As of 7/1/2015, user age ranges from 1 year to a bit over 6 years. Most of users are 1 to 2.5 years old (based on their first activity on the site).

Booking History:

- **How many users have made the booking? What destination are more popular? How long did it take users to convert after getting on the site? How does the user growth look for the website?**
    - Only 42% of users have made a booking.
    - Most users like to travel within US. Europe is the second place they like to go.
    - Majority of users (75%) that have booked their first destination within 30 days since they got on the website for the first time.
    - Between 2013 and 2014, number of users on Airbnb has grown dramatically.
    - Take 2013 as an example to take a look at the new user acquisition trend, you will see theres seasonality when it comes to the user acquisition. October is peak month for user acquisition while the beginning of the year is the off-peak season for user acquisition for Airbnb.

Correlation Exploration:

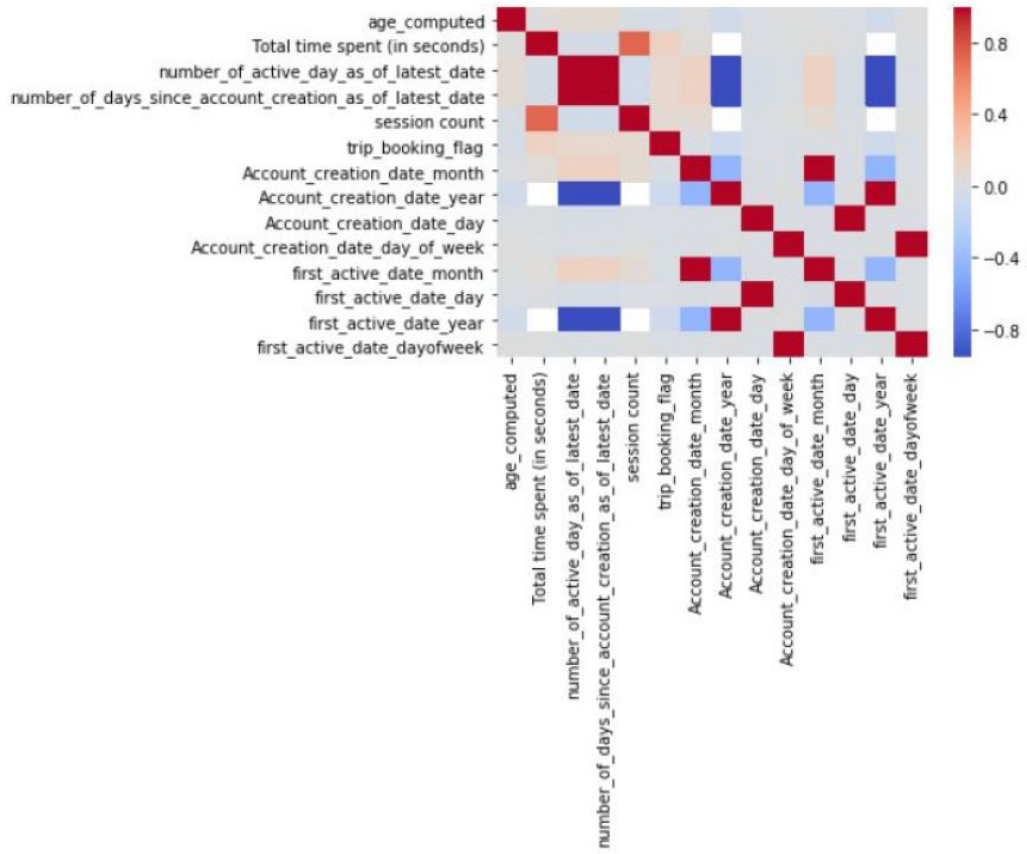
- **Is there any correlation between user acquisition channel and their booking behavior? Is it true the more time users have spent with website the more like users are to make a booking?**
    - Users coming from Web are more likely to make a booking.
    - Users signing up using Mac are more likely to make a booking.
    - Users that have made a booking have longer account age on average than users that have not made a booking as of 7/1/2015.

- Users that have made a booking have done more activities and spent more time with the website on average than users that have not made a booking.

Exploratory Data Analysis Part Two - Variable Statistical Learning

Is there any strong correlation between the label (trip booking flag) and the features?

By looking at the heatmap below, the feature that is most correlated to the trip_booking_flag is the total time spent (in seconds).



Are there strong correlations between pairs of independent variables or *between an independent and a dependent variable?*

By looking at the data, below are the three question/hypothesis I have regarding the relationship between the variables:

- Bookers has stayed with the airbnb website longer than the non bookers (based on the first date active)

- Bookers has more activities on the airbnb website than non bookers (based on the session count)
- Age and gender could be important factors to determine which country users booked

What are the most appropriate tests to use to analyse these relationships?

For each of the hypothesis above, I have come up with the test below:

- Hypothesis 1: Bookers has stayed with the airbnb website longer than the non-bookers (based on the first date active)

  Ho: There's no difference between bookers and non-bookers on their number of days since they got active on the website.
  H1: Bookers stay longer with the website than non-bookers.

  For hypothesis 1, I have used z test to compare the average number of days since the first active date between the bookers and non bookers

  The conclusion is the high z test statistic indicates that the p value is very low and there's significant difference between bookers and non-bookers in number of active days.

- Hypothesis 2: Bookers have more activities on the airbnb website than non-bookers (based on the session count)

  Ho: There's no difference between bookers and non-bookers on their web activities.
  H1: Bookers have more web activities on airbnb than non-bookers on average.

  For hypothesis 2, I have also used z test to compare the average session count between bookers and non-bookers.

  The conclusion is bookers are on average more active than non-bookers on the airbnb website.

- Hypothesis 3: Age and gender could be important factors to predict which country users booked

  For hypothesis 3, I have used data visualization, chi square test and One Way Anova test to determine the interaction between the label and features like user age and user gender.

  By doing the Chi-square test and leveraging data visualization below, we can reach the conclusion that there's a significant relationship between age, gender and booking destination. But the relationship is not that strong given Cramers V is only 0.0635.

In-Depth Analysis

Two step classification

The labels of the dataset are not balanced, more than half of the users have not booked a trip yet. So I decided to do a two-step classification.

For the first step, I'm just going to predict whether or not users have made a booking. For the second step, I'm just going to focus on the data with users that have made a booking and make predictions on what countries users have booked as their first destination country.

First step: Binary Classification problem

Label: Trip booking flag (which indicates if a user has booked or not)

Features:
Here's a screenshot of some of the features included:

```
Index(['id', 'date_account_created', 'timestamp_first_active_cleaned',
       'gender', 'signup_method', 'signup_flow', 'language',
       'affiliate_channel', 'affiliate_provider', 'first_affiliate_tracked',
       'signup_app', 'first_device_type', 'first_browser',
       'country_destination', 'age_computed',
       'Account_creation_before_booking_flag', 'Total time spent (in seconds)',
       'number_of_active_day_as_of_latest_date',
       'number_of_days_since_account_creation_as_of_latest_date',
       'session count', 'trip_booking_flag', 'Account_creation_date_month',
       'Account_creation_date_year', 'Account_creation_date_day',
       'Account_creation_date_day_of_week', 'first_active_date_month',
       'first_active_date_day', 'first_active_date_year',
       'first_active_date_dayofweek', 'age_bucket',
       'Total time spent (in seconds)_fill_null_zero',
       'session count_fill_null_zero', 'Total time spent (in seconds)_bucket',
       'session count_bucket'],
      dtype='object')
```

Other than the features above, the activity counts by event type fields are also included.

For the categorical fields, I have used one-hot encoding to create dummy variables since sklearn cannot take categorical variables directly for modeling.

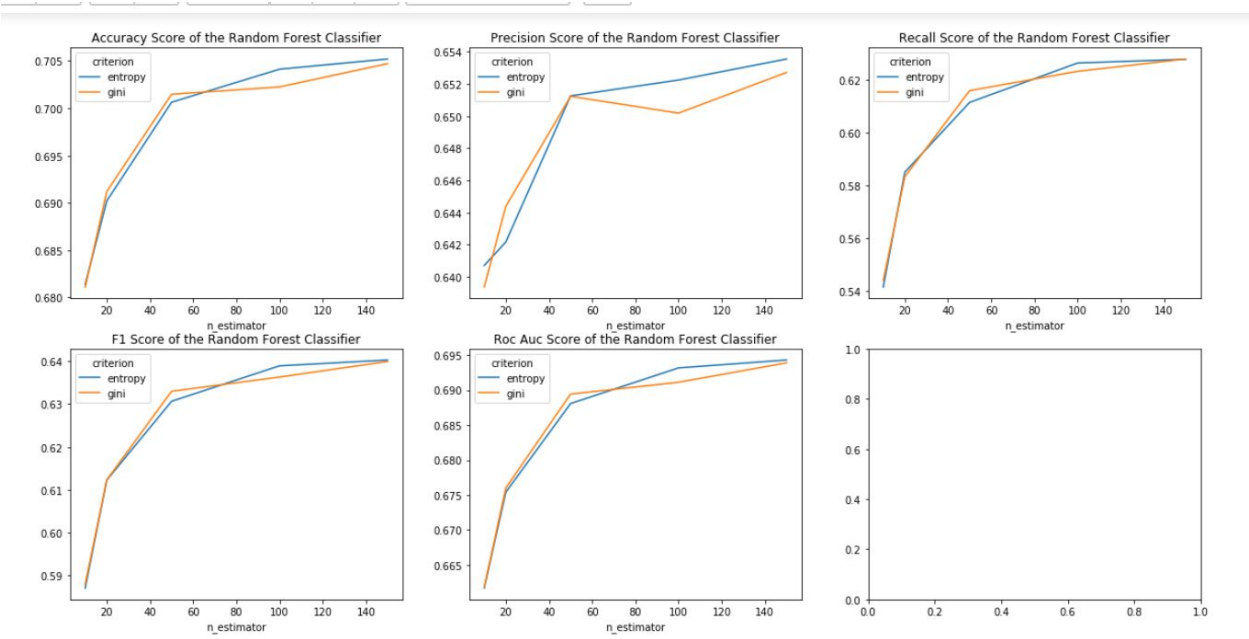
Training and testing sets split for cross validation:
I have split the training and testing set into 70% and 30%. The training set is mainly used for training the model and refining the parameters of the models. The testing set is used for evaluate the performance of the model.

Optimization metric:
The metric to use for optimization include accuracy score, precision, recall and roc auc score.

Algorithm:
I have picked different algorithms to use which include logistic regression, random forest, XG Boost, Gradient Boosting and Adaboost.

Hyperparameter Tuning:
For each of the algorithm, I have done some hyperparameter tuning to find out what is the best parameter to use such as number of estimators, learning rate, etc.

Below is a screenshot of the model performance based on number of estimators in the random forest model. In the hyperparameter tuning, Roc Auc score has been used as the primary KPI to make decision on what parameter to use.



Also, random search and grid search cross validation methodologies have been leveraged to find the best hyper parameters for some bagging and boosting models.

Optimization:
Among all the algorithms I have used, random forest has the best performance on both training and testing set compared with other models.

Below is the performance of the random forest model. As you can see, the AUC score in the training set is 77% and accuracy score is 78%. But the scores on the testing set have a drop-off of 6%, which means the model might have some problems of generalization.

Below is the performance on the training dataset and testing dataset:

```
In [50]:  1  model_evaluation(best_grid, X_train, y_train)
```

```
Model Performance:
accuracy score:  0.7845806980557508
precision score:  0.7665780017732978
recall score:  0.6935125246270757
roc auc score: 0.7715004950116321
```

```
In [51]:  1  model_evaluation(best_grid, X_test, y_test)
```

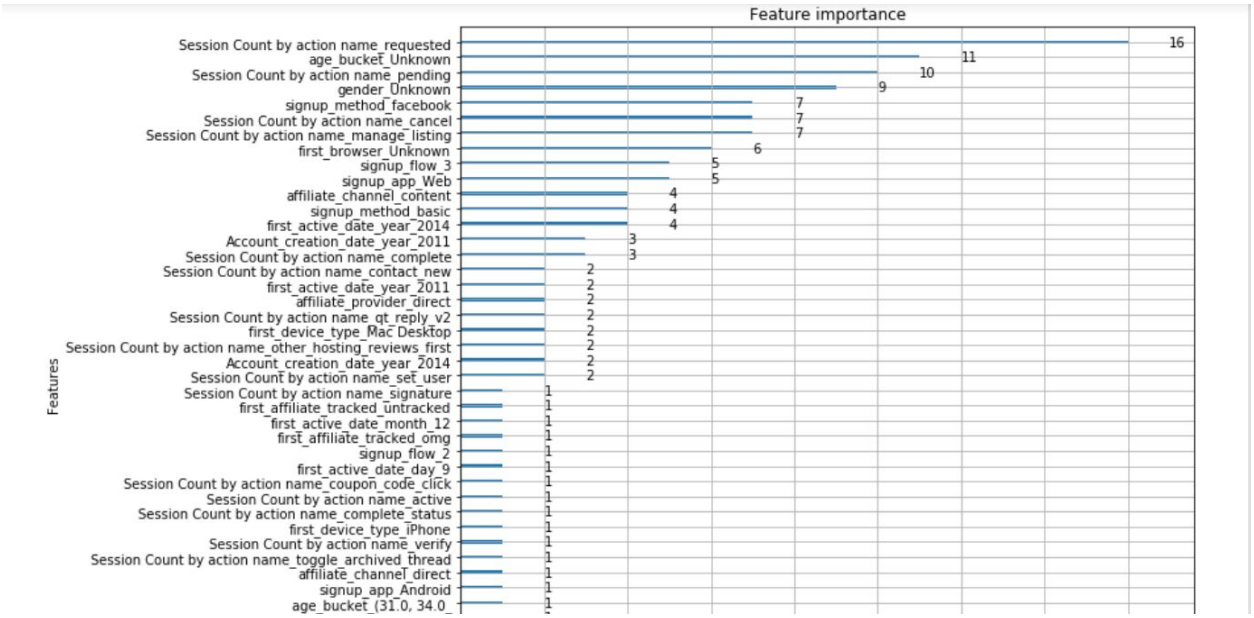
```
Model Performance:
accuracy score:  0.7243447096577733
precision score:  0.6865488463426608
recall score:  0.6268489466606902
roc auc score: 0.7106188540412093
```

Learning and interpretation from the model output:

For each of the model discussed above, I have output the most important factors to classify the users into “booked a trip” and “not booked a trip yet”.

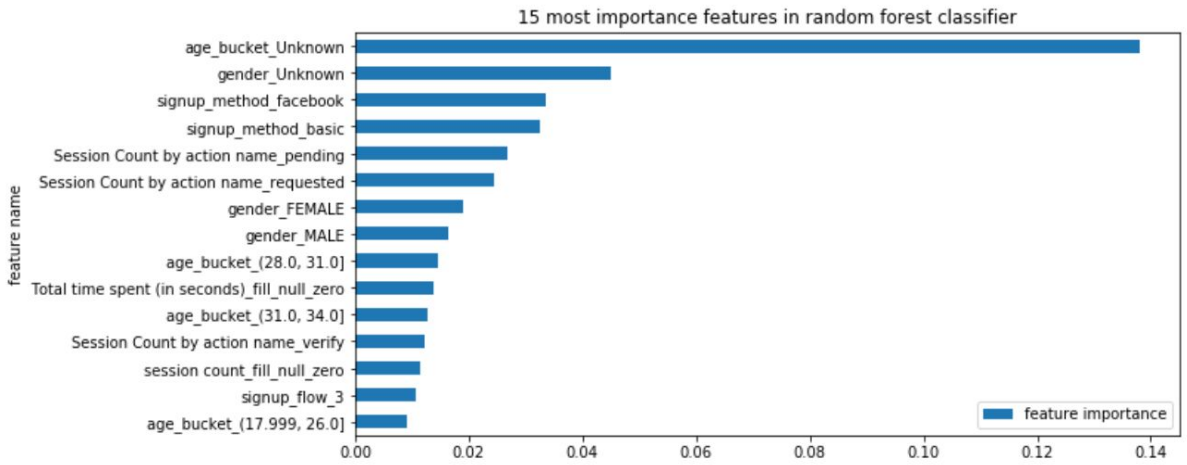
Below is a screenshot of ranking of the important factors from xgboost model:

By looking at the output below, you will find features like session count by action name requested, age bucket unknown, session count by action name pending and gender unknown are important features in the classification.



Below is the screenshot of the important features from the random forest classifier:

Age bucket unknown, gender unknown, sign-up method facebook and sign-up method basic have become important factors in the classification.

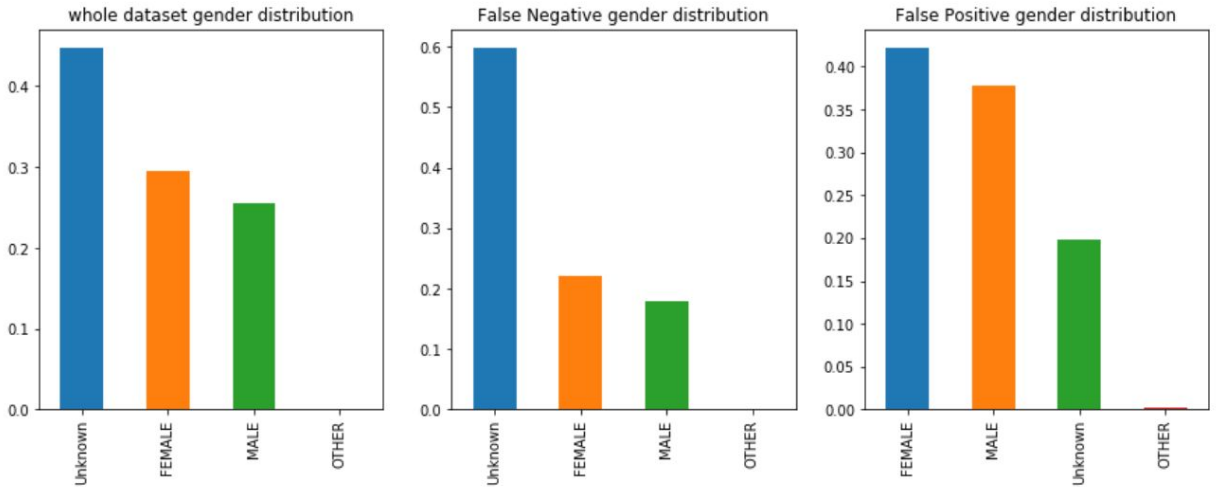


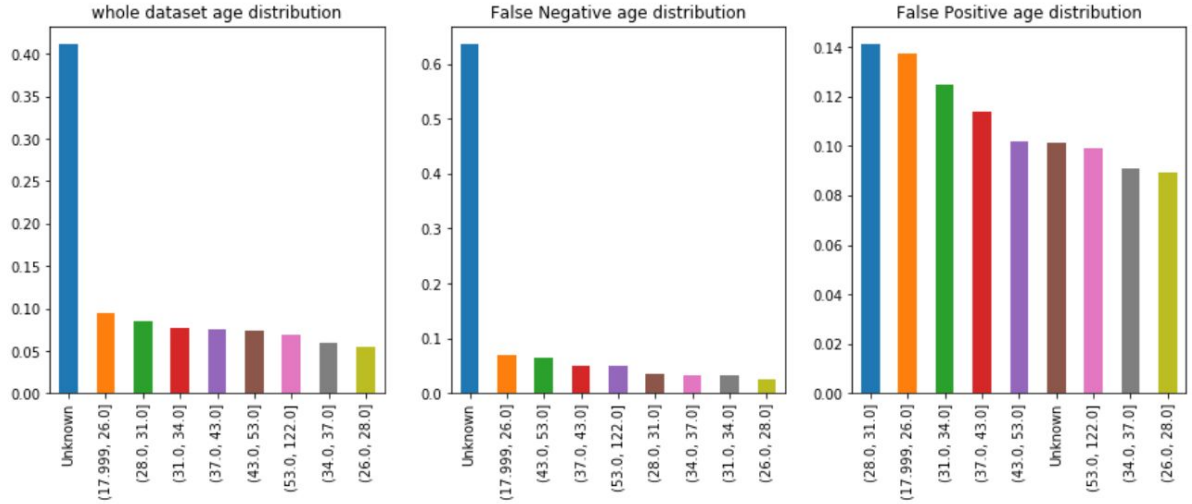
In this case, I think the important features from the xgboost classifier have given me better interpretation since xgboost classifier has given more weight on types of activities users have done on the website than demographic information.

Error analysis:

To understand what would be the reason for misclassification, I have also done error analysis on the training for the random forest classifier.

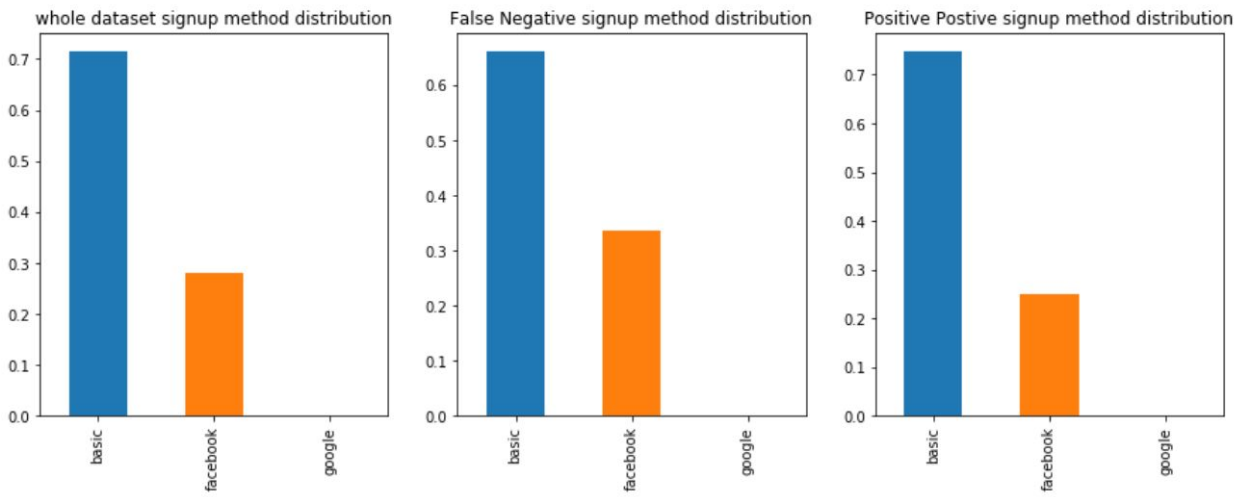
By comparing the actual gender/age distribution of the data with the gender/age distribution of the misclassified examples, you will find that users with unknown gender or age are less likely to classified as bookers.





By comparing the average pending action and requested action session counts between the misclassified examples and actual dataset, it seems that users with more of these types of actions are more likely to be classified as bookers.

For the signup method, users using basic signup method are more likely to be classified as bookers while users that signed up through Facebook are more likely to be classified as non-bookers.

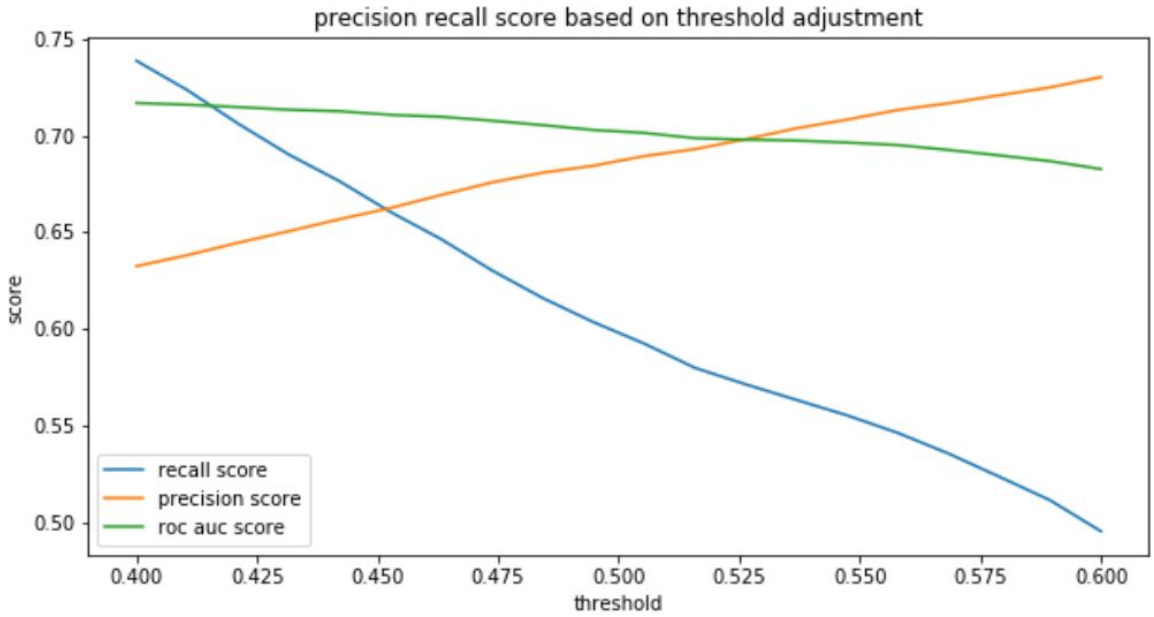


Conclusion from the error analysis:

The misclassification here is mainly due to the imbalanced data on the user profiles. For example, there are many users with unknown gender and unknown age (These users are less likely to be classified as bookers), which has limited the capability of the model to make more accurate classifications. Next step would be to only train on the user data with known gender and age.

Threshold adjustment:

By adjusting the threshold for the probability and plot the precision, recall and roc auc score in the plot, I decided to use 0.44 as threshold instead of the default 0.5 since it returns a better balance between precision and recall.



Second step: Multi-Class Classification problem

Label: Destination Country
Features used: same as the features used for the binary classification problem
Training and testing sets split for cross validation: Same as binary classification

Optimization metric:
The metrics to use for optimization include accuracy score, precision, recall, roc auc score and NDCG score.

Here's how the NDCG score is calculated:

The evaluation metric for this competition is NDCG (Normalized discounted cumulative gain) @k where k=5. NDCG is calculated as:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 (i + 1)},$$

$$nDCG_k = \frac{DCG_k}{IDCG_k},$$

where rel_i is the relevance of the result at position i.

$IDCG_k$ is the maximum possible (ideal) DCG for a given set of queries. All NDCG calculations are relative values on the interval 0.0 to 1.0.

For each new user, you are to make a maximum of 5 predictions on the country of the first booking. The ground truth country is marked with relevance = 1, while the rest have relevance = 0.

For example, if for a particular user the destination is FR, then the predictions become:

[FR] gives a $NDCG = \frac{2^1-1}{log_2(1+1)} = 1.0$

[US, FR] gives a $DCG = \frac{2^0-1}{log_2(1+1)} + \frac{2^1-1}{log_2(2+1)} = \frac{1}{1.58496} = 0.6309$

Algorithm and Optimization:

After trying out random forest, logistic regression and xgboost, I've decided to use xgboost since it's given me the best result (accuracy score and also NDCG score).

On the testing set, the average NDGC score for the XGboost is 0.82 and the accuracy score on the testing set is 0.70.

```
:  1  xgboost_evaluation_final_df_test_set.Score.mean()
:  0.8231714239341013
```

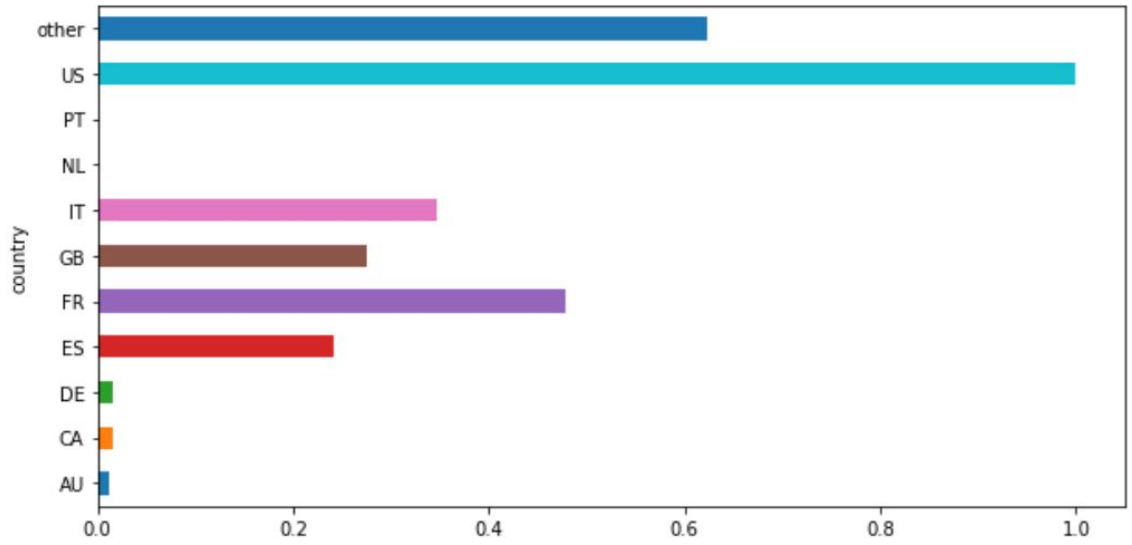
```
:  1  accuracy_score(y_test, model_xgboost.predict(X_test))
:  0.7005961084242492
```

Below are the findings based on the confusion matrix and NDCG score by labels plot:

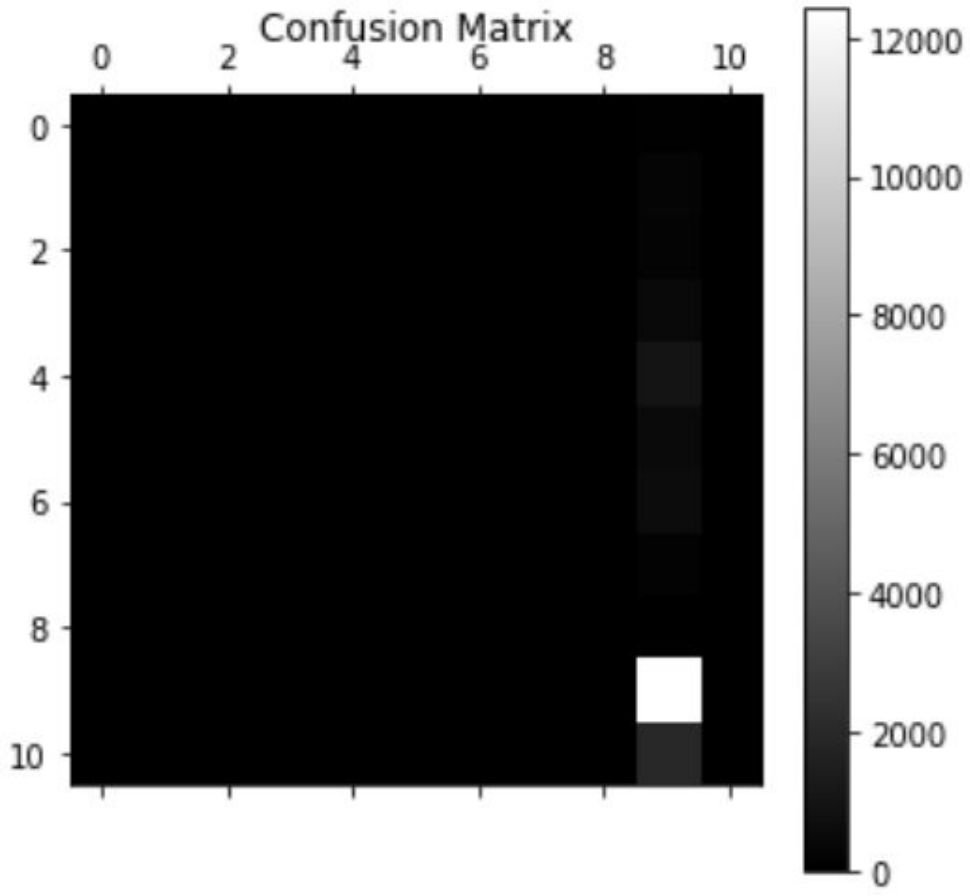
- In the training dataset, the model classifies most of the users into "US".

- The prediction model is doing very poorly for countries like AU, CA, PT, NT, DE, ES and GB.
- The model is performing well in predicting the users who want to go to US as their first destination.
- But the model is also generating errors by misclassifying some of the users who booked other countries into US

Below is the NDCG score by destination countries:

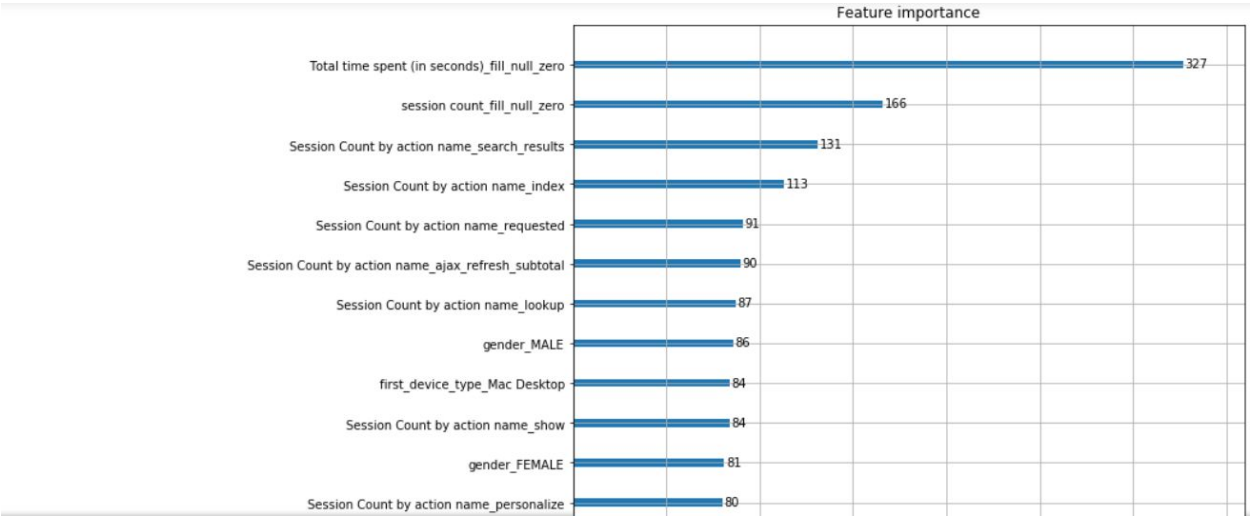


Below is the confusion matrix:



Learning and Interpretations from the model:

Below are the important features in classifying the first destination countries for the users who have already made a book. Some of the important features include total time spent, session count, search result session count, index session count and requested session count.



Next step for the project:

For binary classification, the misclassification is mainly due to the imbalanced data on the user profiles. For example, there are many users with unknown gender and unknown age (These users are less likely to be classified as bookers), which has limited the capability of the model to make more accurate classifications. Next step would be to only train on the user data with known information.

For multi-class classification, tuning on the parameters and increasing the number of estimators would be quite expensive. So I didn't try it. The next step would be to leverage randomized search and grid search for hyperparameter tuning for the xgboost model.