Applied Data Science I590-33626

Final Project Report

Name - Surya Prakash Sekar

Email - sursekar@iu.edu

Table of Contents

1. Introduction 2
2. Data 2
   2.1. Exploration 2
   2.2. Cleaning 3
3. Method 3
4. Results 4
5. Discussion 5
6. Conclusion 5
   6.1. Future Analysis 5
7. Appendix 6

Introduction

New users on Airbnb can book a place to stay in more than 190 countries around the world. The objective of this project is to predict in which country a new user will make their first booking at. By accurately predicting where a new user will book their first travel experience, Airbnb can

- Share more personalized content across their community, which can engage a new user in making their booking with respect to their favourite destinations
- Decrease the average time with respect to the first booking of a new user and making it as hassle free as possible for them
- Better forecast demand which can help Airbnb in allocating their resources in an efficient manner
- Focus better on customized marketing campaign that enables Airbnb to target customers with respect to their predicted choice of country of stay

Data

- **Exploration**

The dataset consisting of the age and gender buckets is examined first. I have reduced the dimension of the age variable by categorizing them as,

- below 25
- 25 to 49
- 50 to 74
- above 75

This kind of pre-processing is done to better understand the trend and data with respect to the different age groups (junk values have been removed, for example – age values <5). With the help of different charts, I was able to draw a few conclusions regarding the data.

- Firstly, a trendline is portrayed to represent the split of the population across the different destination countries with respect to the different age groups. From the graph (Fig.1.) we could make several observations like,
  - ✓ People belonging to age groups 25 to 49 and 50 to 74 have been more active in making their bookings when compared to their counterparts
  - ✓ The most booked country of destinations with respect to all the age groups in descending order are US, DE and so on
  - ✓ People belonging to the age group below 25, more than 80% of the population have booked their country of destination as US

- Secondly, a bar graph (Fig.2.) is constructed to represent the gender wise split of the population across the different age groups. From this graph we can observe that there isn't, much difference in the male to female ratio across the age groups also most of the population belong to the age groups 25 to 49 and 50 t0 74

  Now the training data is examined for trends and patterns,

- Firstly, a horizontal bar graph (Fig.3.) is created to analyse the population split acquired across the different affiliation channels. This could give us some marketing insights and also help Airbnb in analysing the performance of the different paid marketing platforms and to determine the most effective in order to use it effectively to acquire more customers

- Secondly, a stacked bar graph (Fig.4.) is created to examine the percentage split across the different destination countries with respect to language of the user. The following observations were noticed,
  - ✓ The top 3 destinations are NDF, US and Other irrespective of language (except for languages like el, no, fr and fi where there are alternatives for other)

- **Cleaning**

  - ✓ The age data had a lot of missing and inconsistent values, I have used a logic to replace both the missing values and inconsistent values (values that are <5 and >100) by randomly assigning values between the mean and the standard deviation of the age data

  - ✓ The individual components of timestamp first active and account created data was extracted as separated variables to perform deeper analysis with respect to these values and their effect on the target variable

Method

Random Forest classifier

Random forest classifier is an Ensemble algorithm. It creates a set of decision trees from randomly selected subsets of the training set and then aggregates the votes from different decision trees to decide upon the final class of the target object. The reason I have chose random forest classifier for this problem is because,

- The problem at hand is a **12-class classification problem**
- There are other classification algorithms as well (example – decision trees, Naïve Bayes classifier and so on) but I decided to use an ensemble method as it uses the techniques from more than one algorithm to arrive at an optimal decision

- When I had tried classifying the target variable by using decision trees by itself, the accuracy was very low. Therefore, I wanted to try to classify the target variable with the help of many decision trees (which would improve the accuracy) and hence I chose the random forest classifier to achieve it
- Some of the main parameters of the random forest model in R which are analysed and used in the project are ntree, nodesize, importance, ncores, mtry and nfeatures. Some of the pruning techniques used to tweak the parameters in this model are,
    - ✓ **Number of trees** – one of the evaluation criteria for decision is to monitor the error rate while building the forest and look out for convergence to occur
    - ✓ **Tree depth** - there are many possible ways to control the depth of the trees (limit the maximum depth, limit the number of nodes, limit the number of objects required to split and so on). Finally, we can use the fully developed trees to compute performance of shorter trees as these are a "subset" of the fully developed ones
    - ✓ **How many features to test at each node** – By cross validating across a wide range of values we obtain a performance curve and would be able to identify a maximum indicating the best value for the parameter

Results

- By conducting some analysis on the submission file with the help of a 3D Clustered bar graph (Fig.5.), it is observed that almost 80% of the predicted destination country is classified as NDF (meaning there wasn't a booking) and the remaining 20% is mostly classified as the US
- The model has an accuracy of **63%** in its prediction
- The number of trees used were **750**
- A total of **7** variables were tried at each split
- Using the command print(importance(m,type=2)) we can retrieve the importance of each of the predictors and from this table (Table.1.) we can observe that the major factors involved in deciding a user's first destination country according to our model are,
    - ✓ Age
    - ✓ gender
    - ✓ affiliate channel

Discussion

- **Findings**

  - ✓ It is evident from the result that most of the predicted destinations were NDF (booking was never made) and the US, this factor may be because we are dealing with a dataset in which all the customers are from the USA and there are more local bookings rather than international bookings by new customers in general, we will have to do more analysis on the consumer behaviour to find out more about this phenomenon

  - ✓ The fact that we are trying to predict the destination country for the new customers with first activities after 07/01/2014 based on a training data consisting of old customer may also be the reason for the result because trends can always fluctuate with time from season to season when it comes to travel

Conclusion

Therefore, with the help of this model and its predictions on the destination countries for Airbnb's new customer base, Airbnb would be able to better forecast its demands accordingly, use its Email campaigns to target the right customers with appropriate content and in reducing the booking time with the help of personalized webpages for the respective target users.

We have also found out that most of the customers have not made any bookings, therefore we can use this finding, to figure out why this has occurred by either performing webpage analysis (sessions data) to understand under which process flow users have started dropping out from the booking process, so that we can understand the bigger picture and develop a solid story.

- **Future Analysis**

  - ✓ To make the model more reliable and accurate we can better approximate some of the important variables from our data, in which there are missing or inconsistent values. As in, we can build separate models to predict the correct age, language, gender and so on of the customers rather than approximating them using other techniques. This would improve the model in better approximating the decision on the decision country.

  - ✓ We can deep dive and do more analysis on the web metrics that are available in the sessions data to figure out the factors which would contribute more towards the prediction of the target variable and incorporate them in the model in such a way that the accuracy would be boosted

Appendix

Fig.1.

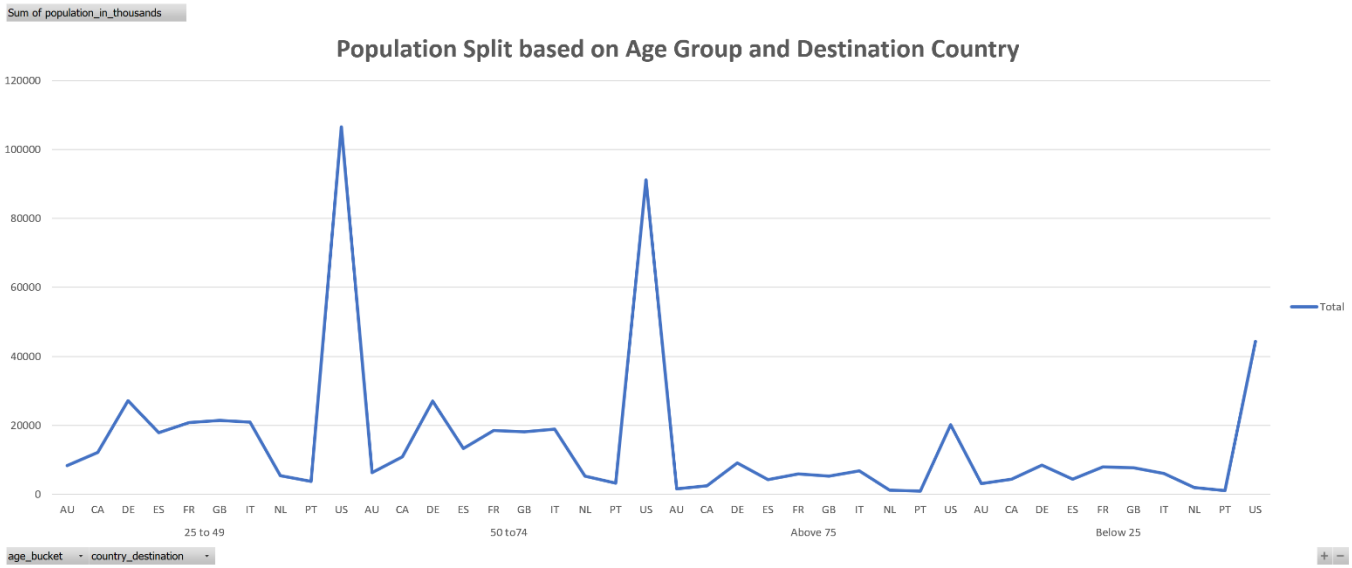

Fig.2.

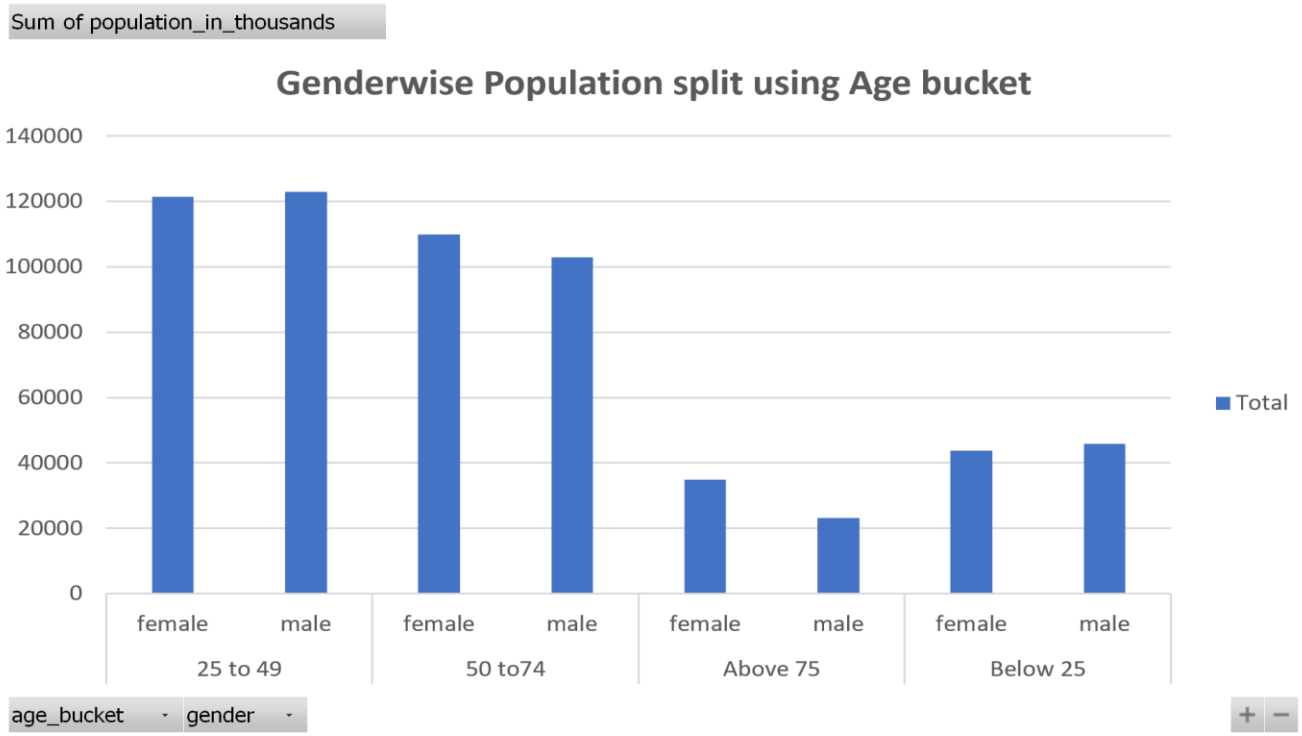

Fig.3.

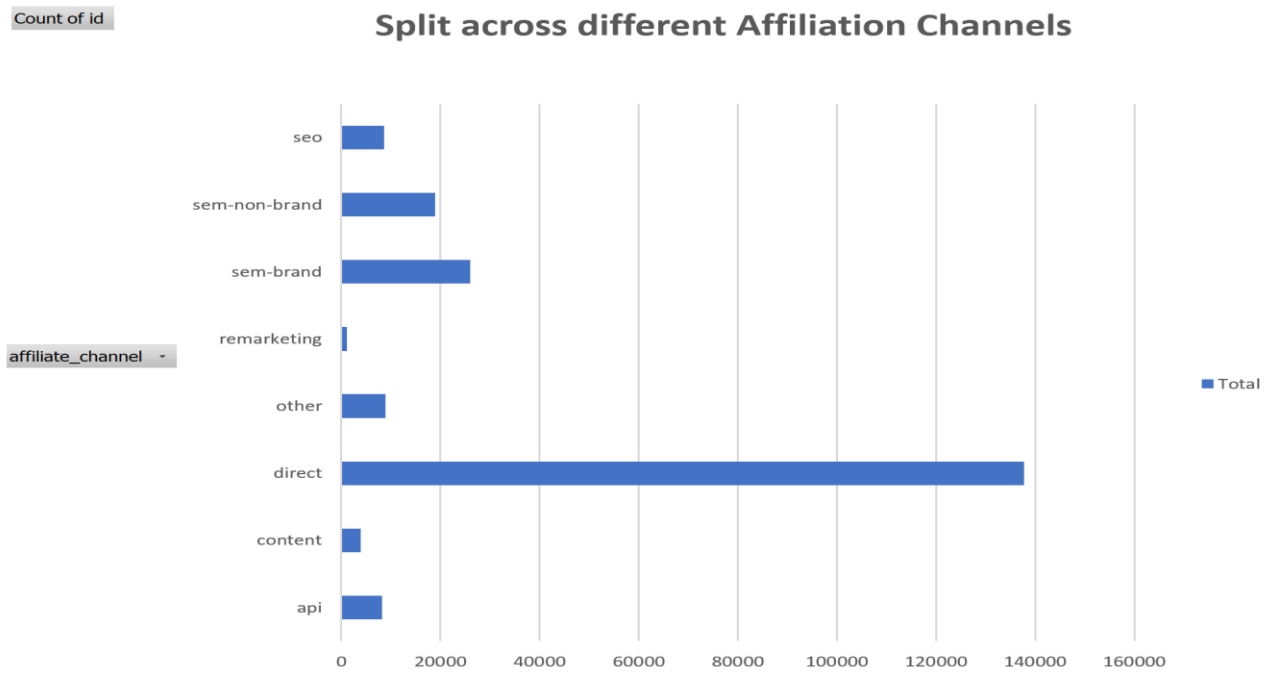
Count of id
Split across different Affiliation Channels
seo
sem-non-brand
sem-brand
remarketing
affiliate_channel
other
direct
content
api
Total
0
20000
40000
60000
80000
100000
120000
140000
160000

Fig.4.

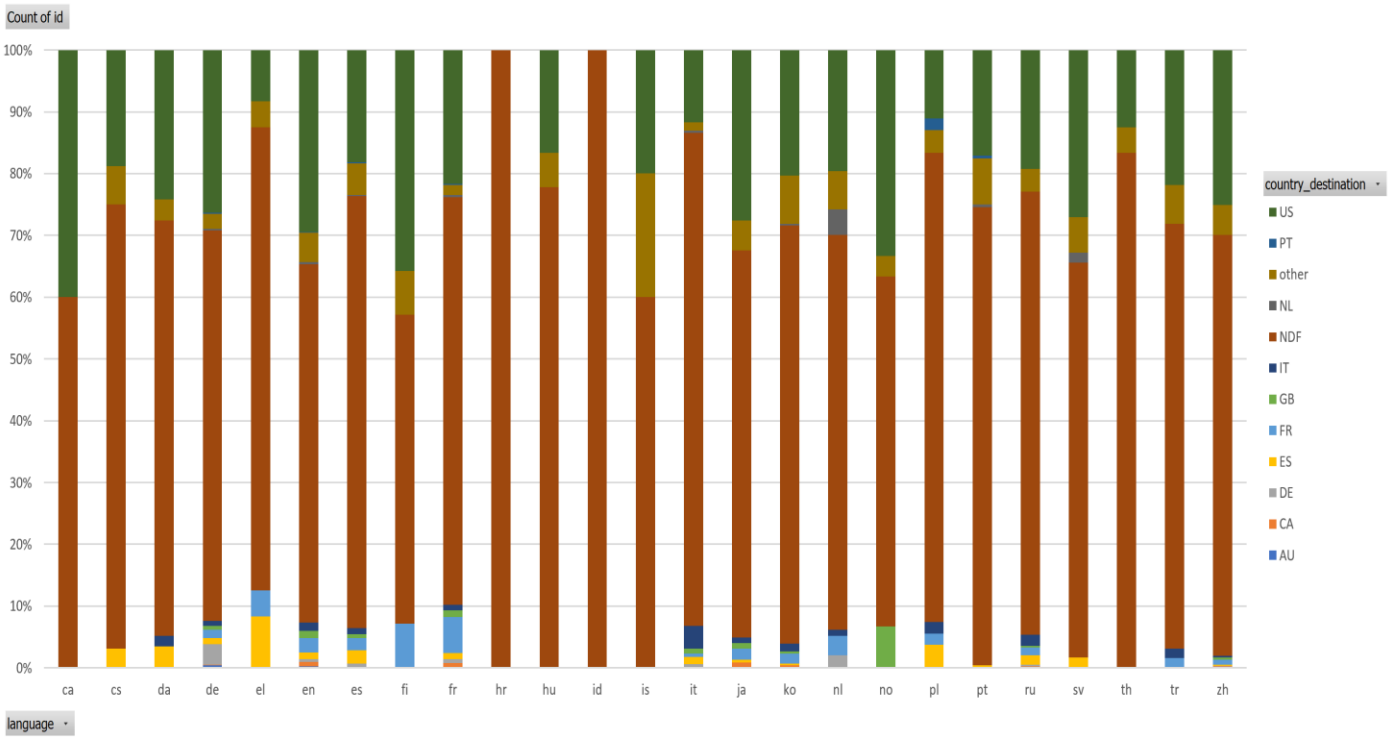
Count of id
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
ca
cs
da
de
el
en
es
fi
fr
hr
hu
id
is
it
ja
ko
nl
no
pl
pt
ru
sv
th
tr
zh
language
country_destination
US
PT
other
NL
NDF
IT
GB
FR
ES
DE
CA
AU

Fig.5.

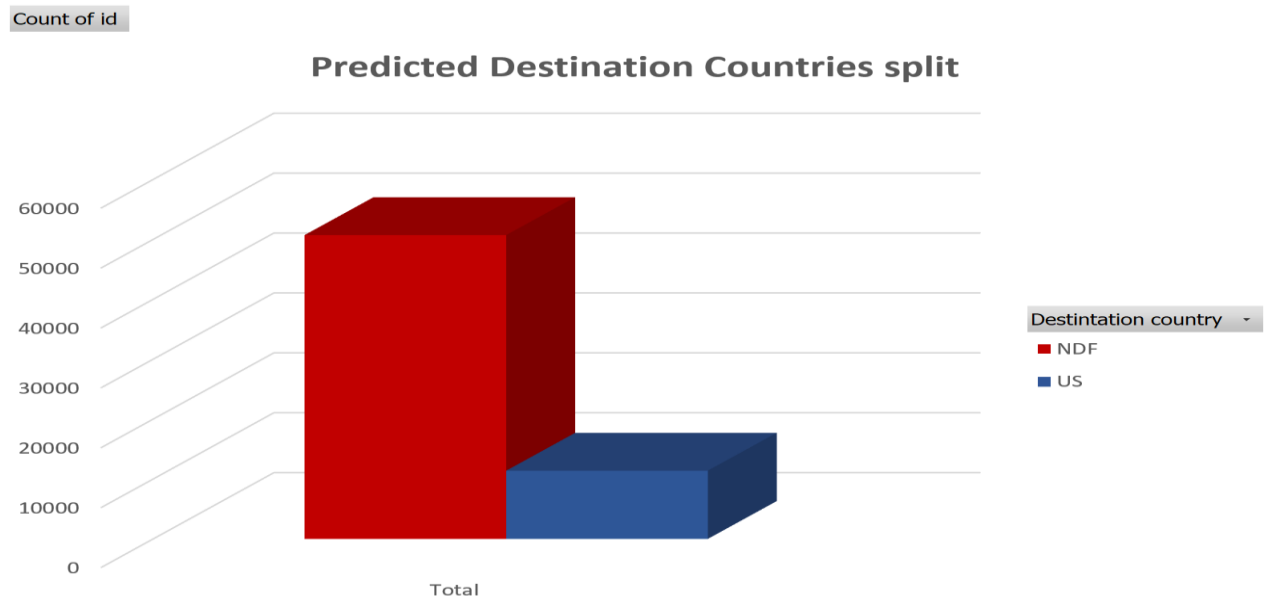

Fig.6.

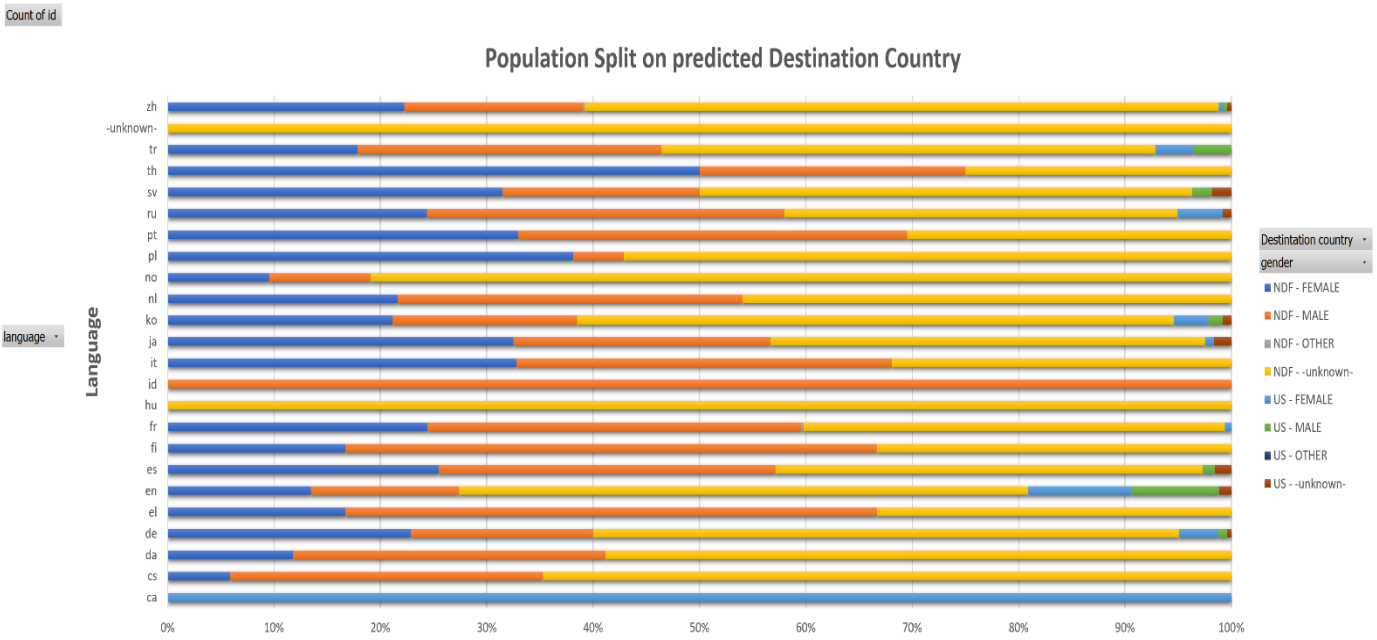

Table.1.

Variable	Importance
Age	6221
Gender	3704
Language	1070
Affiliate Channel	2105
Account Created Year	1828
Signup Flow	1162
Signup App	706

R Code

```
#Libraries
install.packages("c50")
library(C50)                    # decision tree
install.packages("lubridate")
library(lubridate) # date functions
install.packages("dplyr")
library(dplyr)                  # data selection and filter
install.packages("caret")
library(caret)                  # creating data partition
library(randomForest)           # Random forest

#Loading the data set
getwd()
setwd("C:\\Users\\Surya\\Desktop\\IU\\Fall 17\\Applied Data Science\\Assignment\\Final Project")
train_data<-read.csv("train_users_2.csv",header=TRUE)
test_data<-read.csv("test_users.csv",header=TRUE)

#Data Cleaning

#check for null values columnwise
apply(apply(train_data,2,is.na),2,sum)
apply(apply(test_data,2,is.na),2,sum)

# creating arbitrary Ids in the two datasets
train_data$IDA<-1:nrow(train_data)              # Create an arbitrary index for separation
test_data$IDA<-300000 : 300000 + nrow(test_data)
```

```
#  We need to bind the train and test data together, process the data and then separate

# From he training data we need to tremove the variable 'country_destination' and store it somewhere else

labels<-train_data[,c(1,16)]                    # Extracted the country_destination column in the training data

train_data$country_destination<-NULL            # removing 'countrty_destination' column in the training data before binding

all_data<-rbind(train_data,test_data)                   # Binding train and test data

all_data$date_first_booking<-NULL               # Removing date_first_booking col

# Convert date_account_created to date and then extract individual components year, month, day and day of week

all_data$date_account_created<-ymd(all_data$date_account_created)

all_data$account_created_year<-year(all_data$date_account_created)

all_data$account_created_day<-day(all_data$date_account_created)

all_data$account_created_month<-month(all_data$date_account_created)

all_data$account_created_wday<-wday(all_data$date_account_created)

all_data$date_account_created<-NULL

# Applying above technique for 'timestamp_first_active'

all_data$timestamp_first_active<-ymd_hms(all_data$timestamp_first_active)

all_data$ts_first_active_year<-year(all_data$timestamp_first_active)

all_data$ts_first_active_month<-month(all_data$timestamp_first_active)

all_data$ts_first_active_day<-day(all_data$timestamp_first_active)

all_data$ts_first_active_wday<-wday(all_data$timestamp_first_active)

all_data$timestamp_first_active<-NULL

# One factor level is missing in first_affiliate_tracked
```

```
str(all_data)  # First factor level of first_affiliate_tracked is "", assign something to it
levels(all_data$first_affiliate_tracked)[1]<-"missing"

# cleaning Age data

#Setting all values above 100 and below 5 to NA's
all_data$age[all_data$age >= 100]<- NA
all_data$age[all_data$age <= 5]<- NA

#Idea is to replace NA's with random values between Mean-SD and Mean+SD
set.seed(10112017)

#Generating random values between Mean-SD and Mean+SD
random=trunc(runif(90418, min=mean(all_data$age,na.rm=TRUE)-
sd(all_data$age,na.rm=TRUE)-1,
max=mean(all_data$age,na.rm=TRUE)+sd(all_data$age,na.rm=TRUE)+1))

#Replacing NA's with random values between Mean-SD and Mean+SD
all_data$age[is.na(all_data$age)]<-random

# Splitting train and test data
X <-all_data %>% filter (IDA < 300000) %>% mutate(IDA = NULL)

# Also merge in X the class variable: labels
X<-merge(X,labels,by='id')
trainindex<-createDataPartition(X$country_destination,p=0.8,list=FALSE)
training<-X[trainindex,]
test<-X[-trainindex,]

#Building a random forest model
```

```
m=randomForest(country_destination~age+gender+language+affiliate_channel+account_created_year+signup_flow+signup_app,data=training,ntree=750,mtry=7,importance=F,nfeatures=F,ncores=C)

#Validating the model with the training data

v_pred1 <- predict(m, training[,-1])

com1<-data.frame(predicted=v_pred1,actual=training$country_destination)

accuracy1<-sum(com1$predicted == com1$actual)/nrow(com1)

#accuracy1

#[1] 0.6256654

#summary(m)

# Performing Classification on testing data

X_test1 <- all_data %>% filter (IDA >= 300000) %>% mutate (IDA = NULL)

y_pred1 <- predict(m, X_test1[,-1])

# submit results

submit1<-data.frame(id=X_test1$id,country=y_pred)

write.csv(submit1, "submissionrf.csv", quote=FALSE, row.names = FALSE)
```