



AIRBNB CLASSIFICATION - A CLASSIFICATION PROJECT BASED ON R



Priyanka Batavia



Siddharth Mandgi



Chenyu Tian



Introduction

- Airbnb, headquartered in San Francisco, operates a global online marketplace and hospitality service accessible via its websites and mobile apps. Members can use the service to arrange or offer lodging, primarily homestays, or tourism experiences
- Airbnb is an online marketplace which lets people rent out their properties or spare rooms to guests. Airbnb takes 3% commission of every booking from hosts, and between 6% and 12% from guests
- New users on Airbnb can book a place to stay in 34,000+ cities across 190+ countries. 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.

Data



Our dataset is taken from Kaggle



This dataset is taken from a recruiting competition, Airbnb challenges the participants to predict in which country a new user will make his or her first booking



There are 19 variable's in our dataset



For the destination country there are 12 outcomes



The test dataset consists of various attributes like ID, gender, age, date_first_booking, signup_method, signup_flow, language, affiliate_channel, country_destination (which is the target variable to be predicted)

Stages of Our project

Our Project is Divided into Four Stages.

- Data Analysis and Visualization
- Data Preprocessing
- Binary Classification
- Multi-Classification

Data Analysis and Visualization

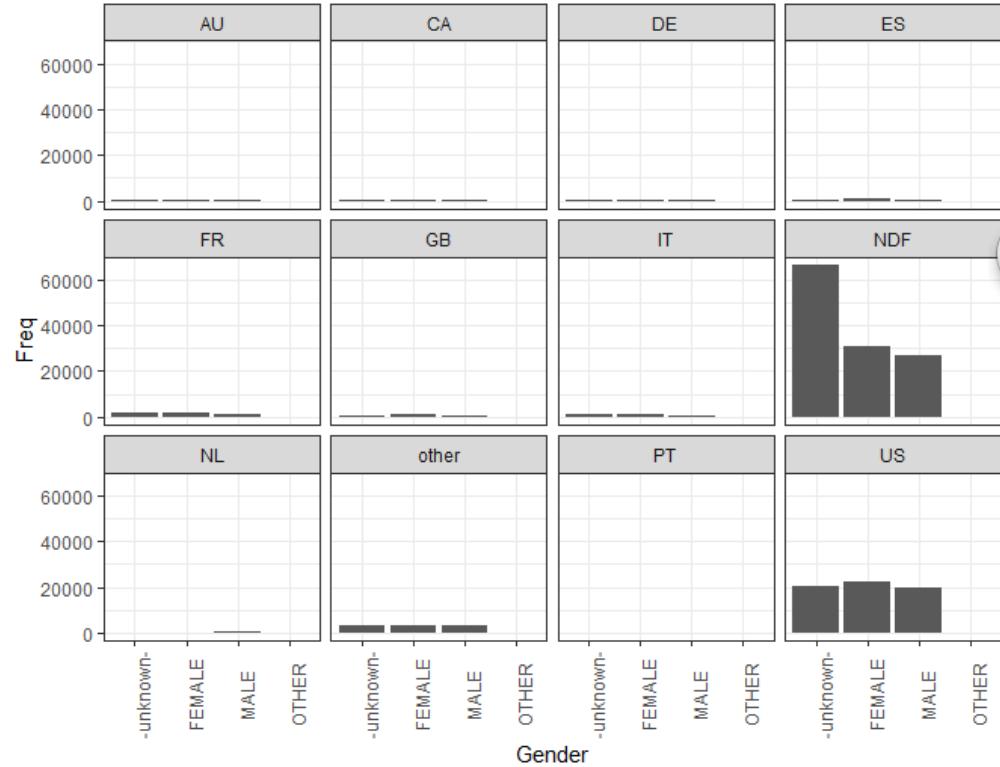
The packages used for our visualizations were tidyverse, ggplot and data.table

The first Visualization gives us the frequency count of gender wise for every country. We can see that the highest count is for the US. (NDF means no destination found and there wasn't a booking)

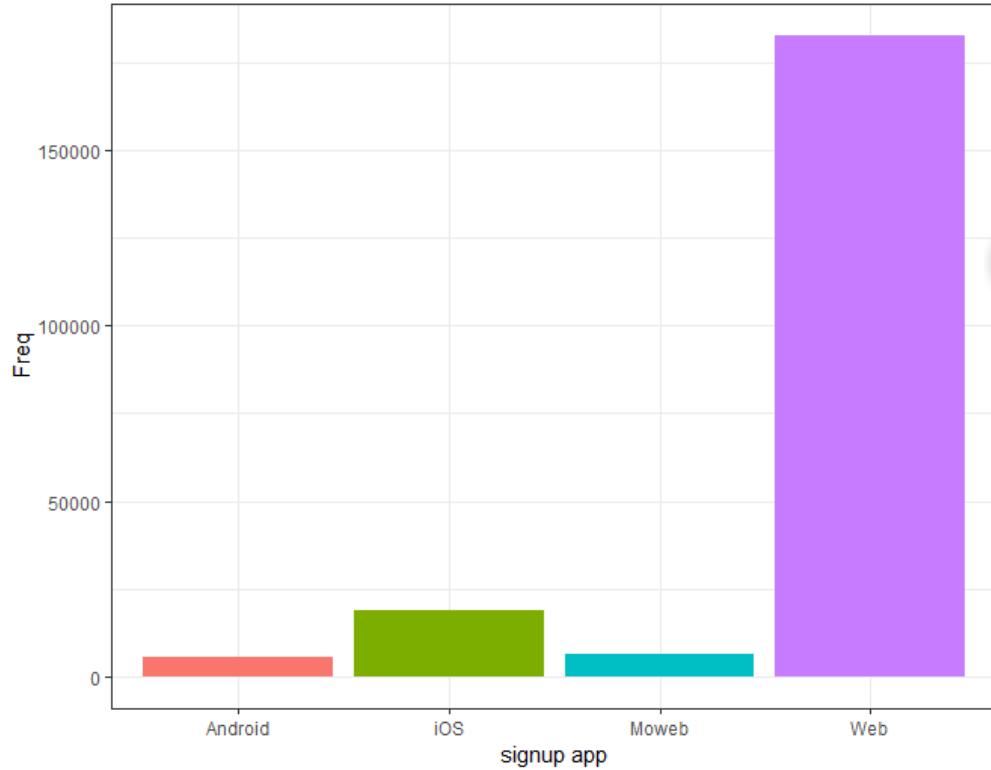
In the second visualization we can see the sign up method every user uses to book an Airbnb. Most of the users use their web browser to book the Airbnb

The last Visualization gives us the distribution of count categorized by age of all the users. The mean age for an average user is between 35-40

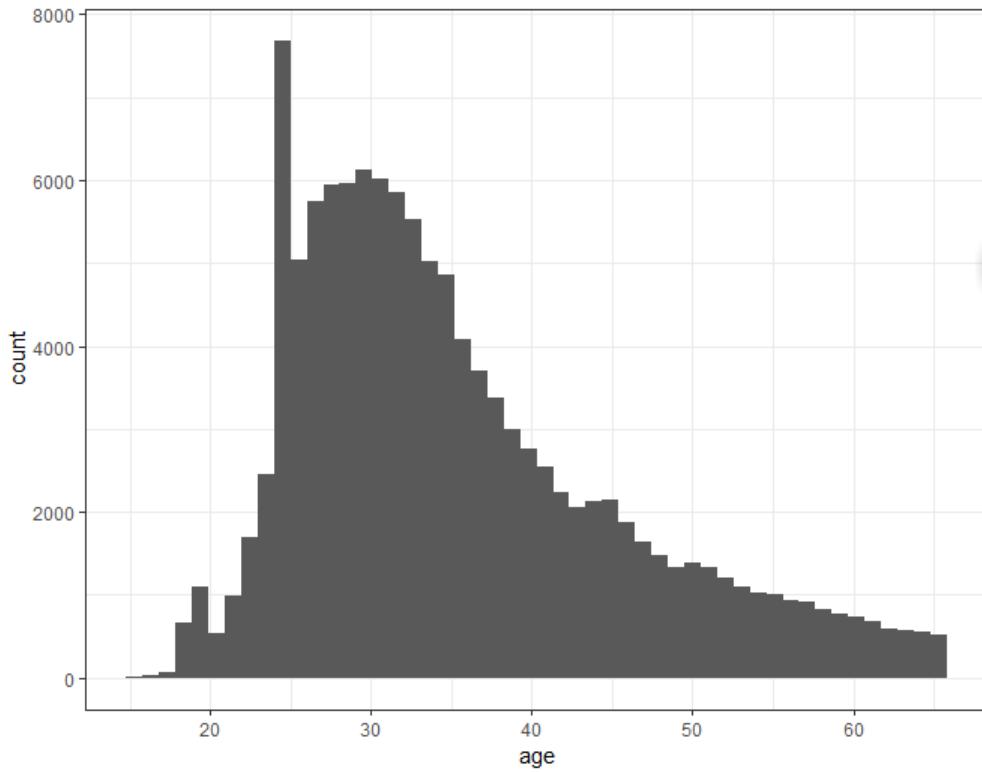
Data Visualization and Analysis



Data Visualization and Analysis



Data Visualization and Analysis



Data Preprocessing.

Removing columns which adds minimal value e.g: -
date_first_booking

Replacing the missing values with mean for numerical columns.

Omitting the missing values which affect the categorical data

We have then split some columns to increase its relevance and meaning e.g :- splitting date_account_created in year, month and day and split timestamp_first_active in year, month and day

ONE HOT ENCODING.

- One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.
- We use one hot encoder to perform “binarization” of the category and include it as a feature to train the model.
- In Our project we have created binary labels for our target variable `country_destination` and split them into several target variables i.e - each country has a label ‘1’ or ‘0’.

ONE HOT ENCODER

```
# one-hot-encoding features
ohe_feats = c('country_destination')
dummies <- dummyVars(~ country_destination, data = df)
df_all_ohe <- as.data.frame(predict(dummies, newdata = df))
df_combined <- cbind(df[,-c(which(colnames(df) %in% ohe_feats))],df_all_ohe)
```

ONE HOT ENCODER

	country_destination
	NDF
	NDF
	US
	other
	US
	NDF
	FR
	NDF
	NDF
	CA



Models used for Analysis

NAIVE BAYES

KNN

ARTIFICIAL NEURAL NETWORKS (ANN)

RANDOM FOREST

C50

H-CLUSTERING AND K-Means

MULTI-CLASSIFICATION

NAIVE BAYES

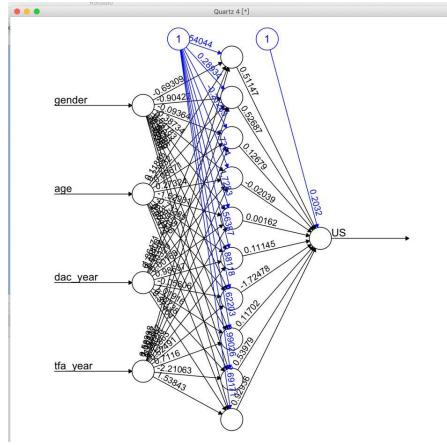
```
US
NBayes      0      1
0 26514  9822
1 3805   2550
> NB_wrong<-sum(category!=test$US )
> NB_error_rate<-NB_wrong/length(category)
> NB_error_rate
[1] 0.3192008
> accuracy <- (1-NB_error_rate)*100
> accuracy
[1] 68.07992
>
```

- We have used 100% of our data while applying this model

KNN

```
kknn      0      1
 0 8923 3069
 1 1266  973
> knn_error_rate=sum(fit!=test$US)/length(test$US)
> print(knn_error_rate)
[1] 0.3046167
> accuracy <- (1-knn_error_rate)*100
> accuracy
[1] 69.53833
> |
```

- We have used 33% of our data while applying KNN
- The main attributes considered were: Gender, Age, dae_year,tfa_year

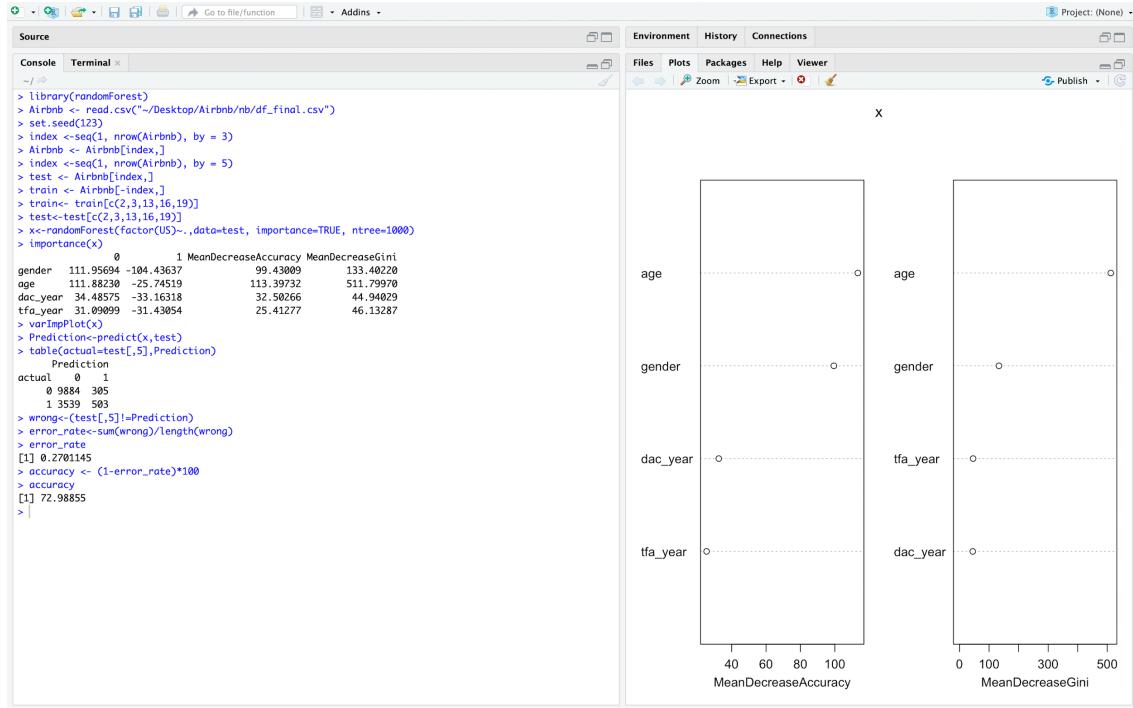


```
35.1 [Top Level] 
Console Terminal 
~/ 
> #Confusion Matrix 
> table(Actual=test$US,predition=ann_cat) 
predition 
Actual      0      10189      1      4042 
0          10189 
1          4042 
> wrong<- (test$US!=ann_cat) 
> error_rate<-sum(wrong)/length(wrong) 
> error_rate 
[1] 0.2840278 
> accuracy <- (1-error_rate)*100 
> accuracy 
[1] 71.59722 
> 
```

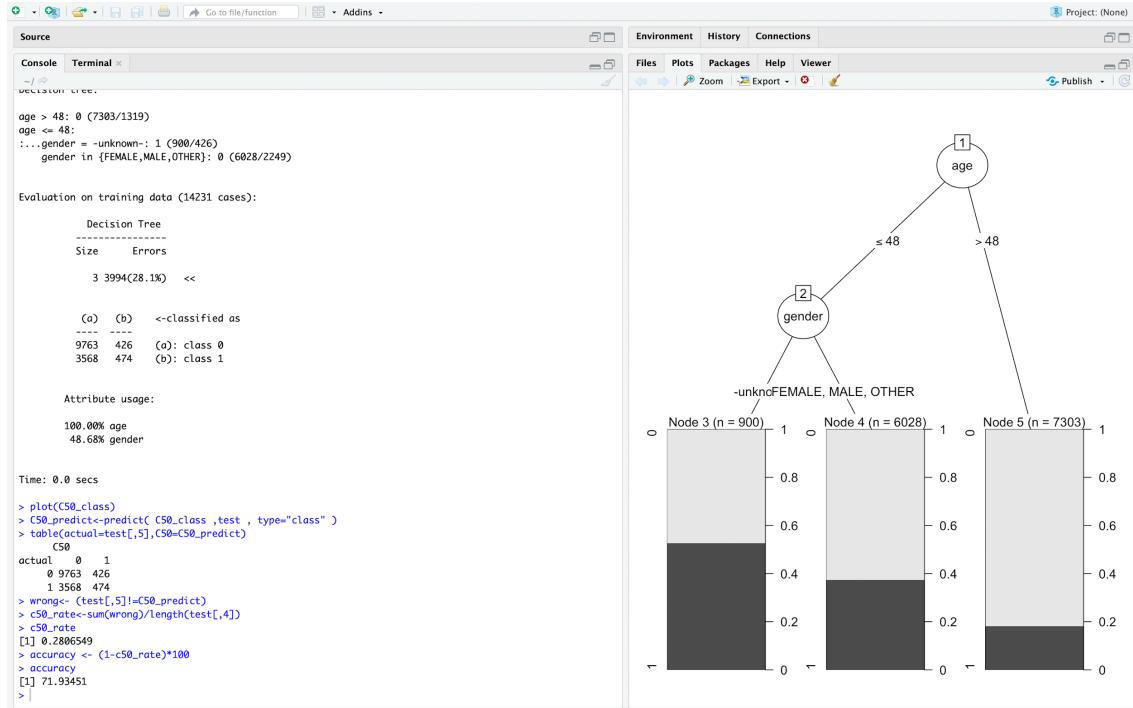
ANN

- We have used 50% of the data while applying this model
- The main attributes considered were: Gender, Age, dae_year,tfa_year

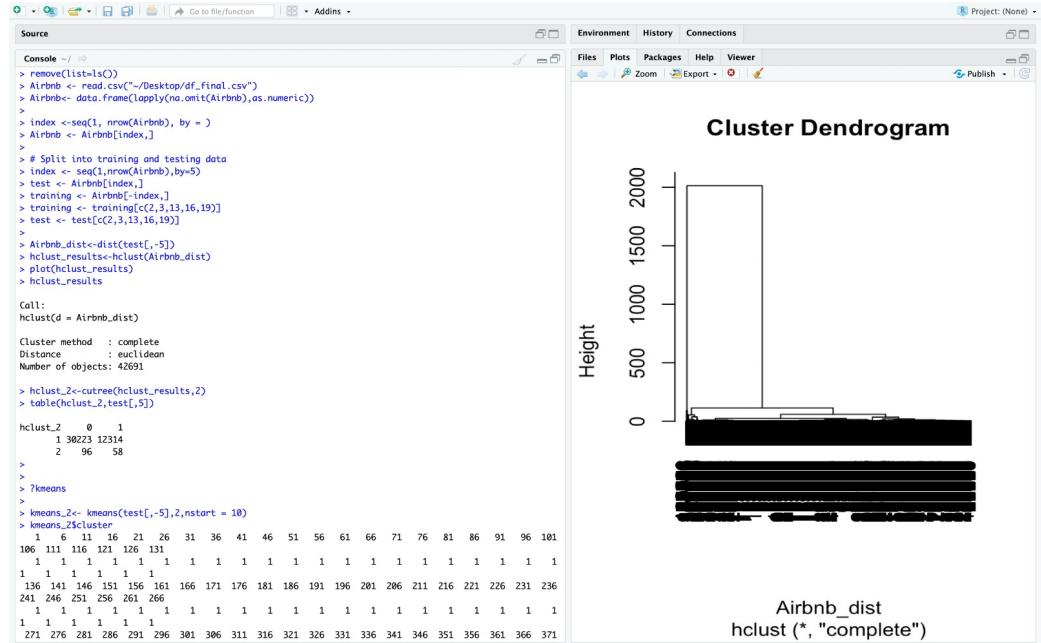
Random Forest



C50

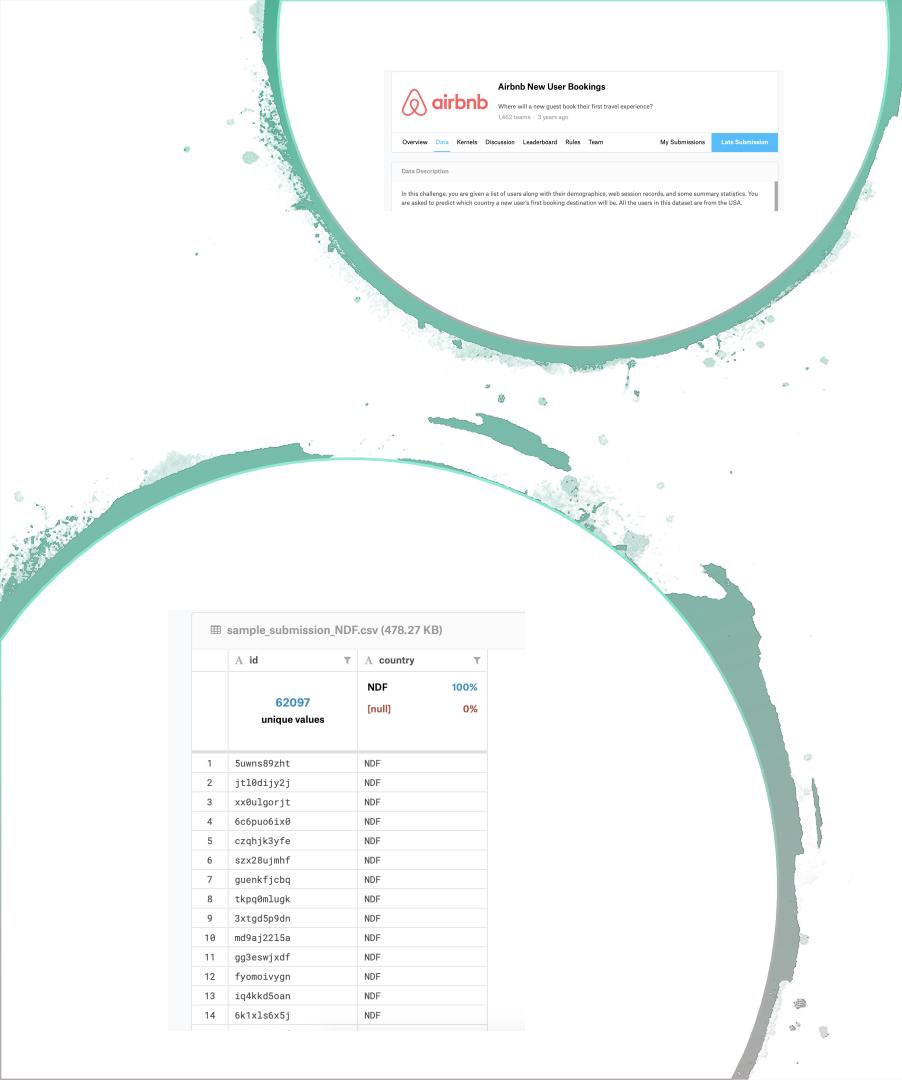


H-Clustering and K Means

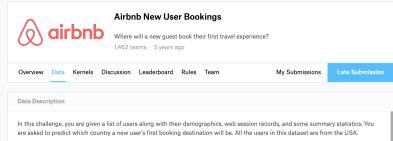


Comparing accuracies of different models

MODELS	ACCURACIES
NAIVE BAYES	68.07992
KNN	69.53833
ANN	71.59772
RANDOM FOREST	72.98855
C50	71.93415



sample_submission_NDF.csv (478.27 KB)			
	A id	T	A country
	62097		NDF
	unique values		100%
		[null]	0%
1	Suwns89zht		NDF
2	jti10dijy2j		NDF
3	xx8ulgorjt		NDF
4	6cdpuo61x0		NDF
5	czphj1k3yfe		NDF
6	szx28ujmhf		NDF
7	guenkfjcbq		NDF
8	tkpq0mlugk		NDF
9	3xtgdp9pdn		NDF
10	md9aj2215a		NDF
11	gg3eswjaxdf		NDF
12	fymomivgn		NDF
13	1qkkkd5oan		NDF
14	6k1xl6x5j		NDF



Multi-Classification

- A classification task with more than two classes; e.g., classify a set of images of fruits which may be oranges, apples, or pears. Multi-class classification makes the assumption that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time.
- The challenge in Kaggle was to classify the destination for each particular user.

Multi- Classification

- We have used the library Xgboost for this purpose
- XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance
- We have used XGBoost because it is way more faster than the existing gradient boosting applications and supports various objective functions including classification and ranking

Multi- Classification

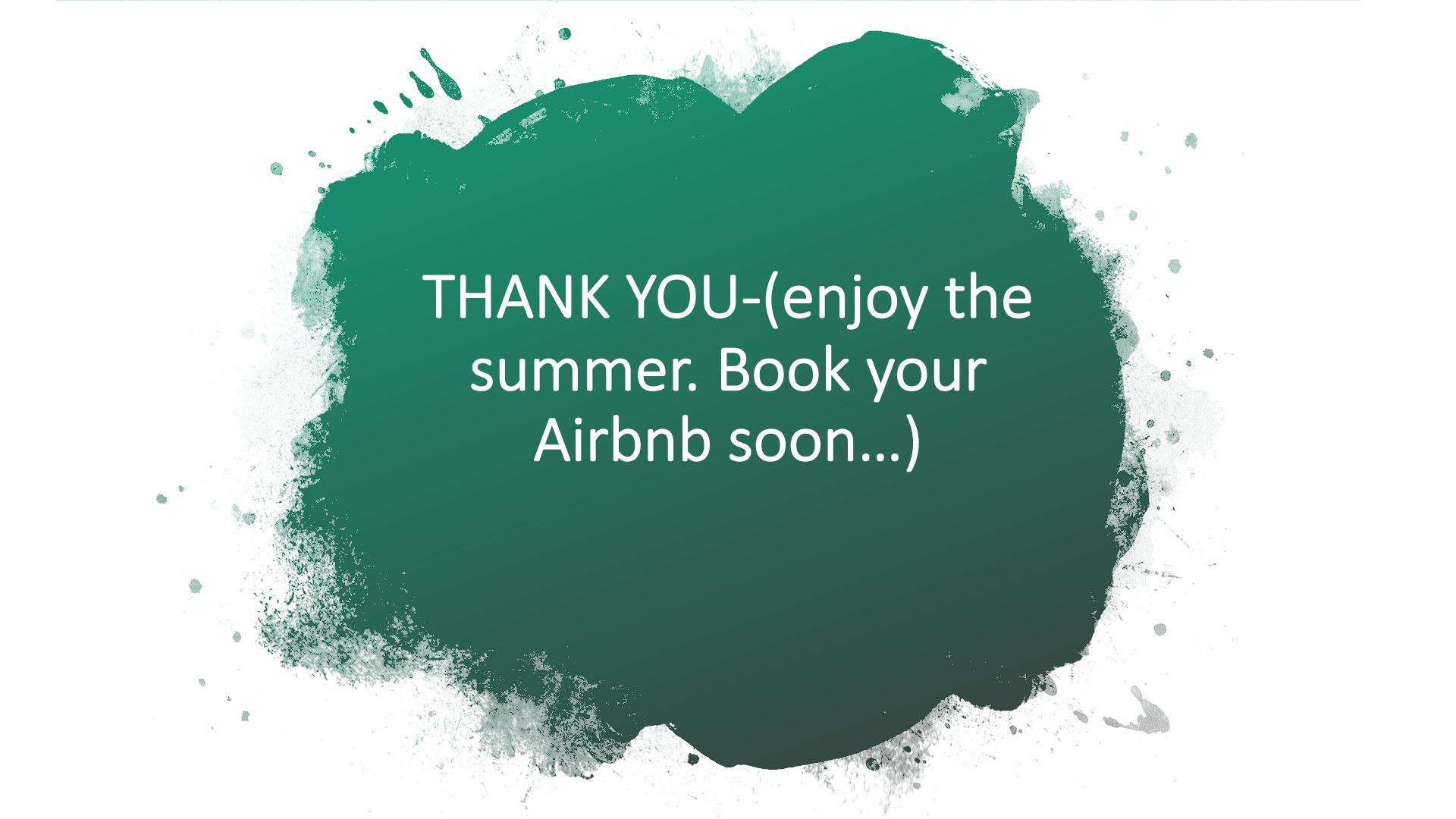
```
xgb <- xgboost(data = data.matrix(X[,-1]),  
                 label = y,  
                 eta = 0.1,  
                 max_depth = 9,  
                 nround=25,  
                 subsample = 0.5,  
                 colsample_bytree = 0.5,  
                 seed = 1,  
                 eval_metric = "merror",  
                 objective = "multi:softprob",  
                 num_class = 12,  
                 nthread = 3
```

```
|  
y_pred <- predict(xgb, data.matrix(X_test[,-1]))
```



Conclusion

- There are more number of females booking an Airbnb
- The Average age people booking ranges from 30 - 40 years
- The most used signup platform is a Web Browser (Safari , Chrome etc.)
- From binary classification we can conclude that Random Forest Classifier works the best.



THANK YOU-(enjoy the
summer. Book your
Airbnb soon...)