STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

THE AIRBNB CLASSIFICATION PROJECT

—

Chenyu Tian

Priyanka Batavia

Siddharth Mandgi

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.

The Data

- Our dataset is taken from Kaggle
- The 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)

,

Data Description

File descriptions

- **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

Details and Main features in our data.

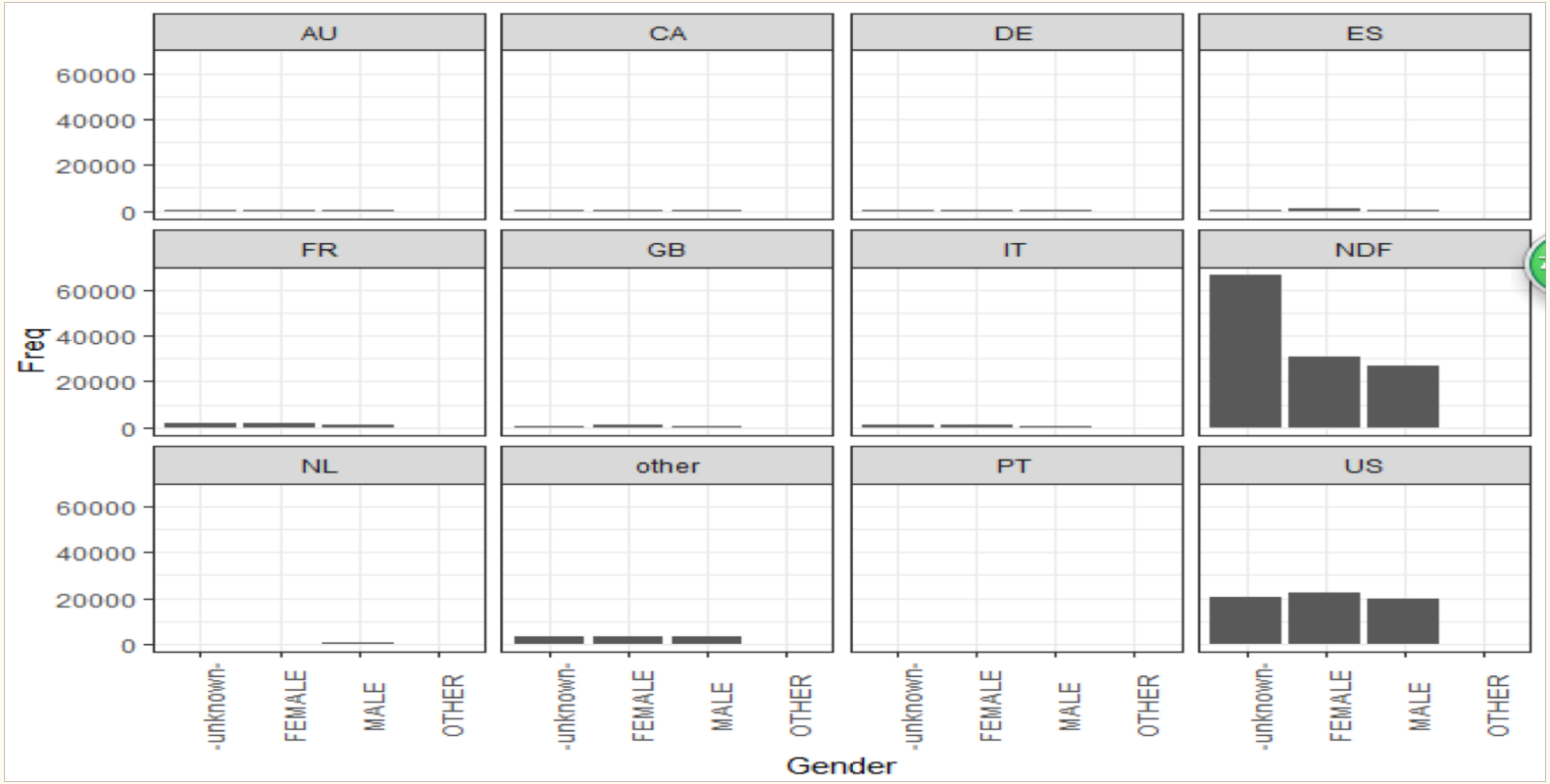
id:	**Value Type:** Char	Gender	**Value Type:** Char	first_device_type	**Value Type:** Char
date_account_created	Int	Age	Int	first_browser	Char
timestamp_first_active:	float	signup_method	Char	affiliate_provider	Char
date_first_booking:	Int	signup_app	Char	affiliate_channel	Char

The Four Stages of Our project

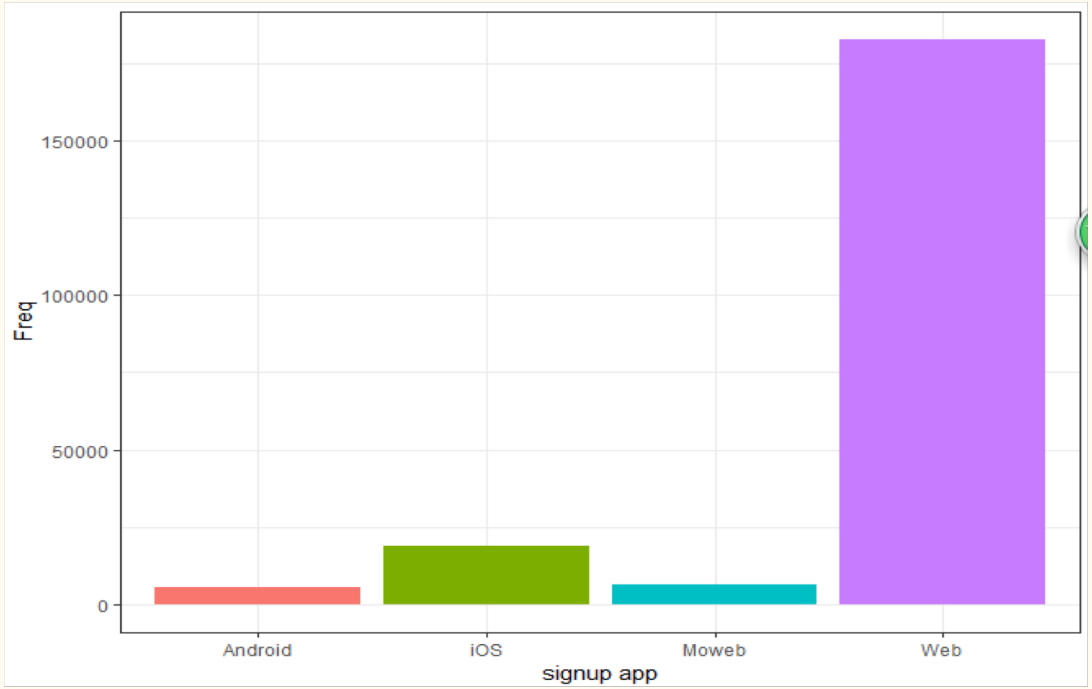
Our Project is Divided into Four Stages.

1. Data Analysis and Visualization
2. Data Preprocessing
3. Binary Classification
4. Multi-Classification

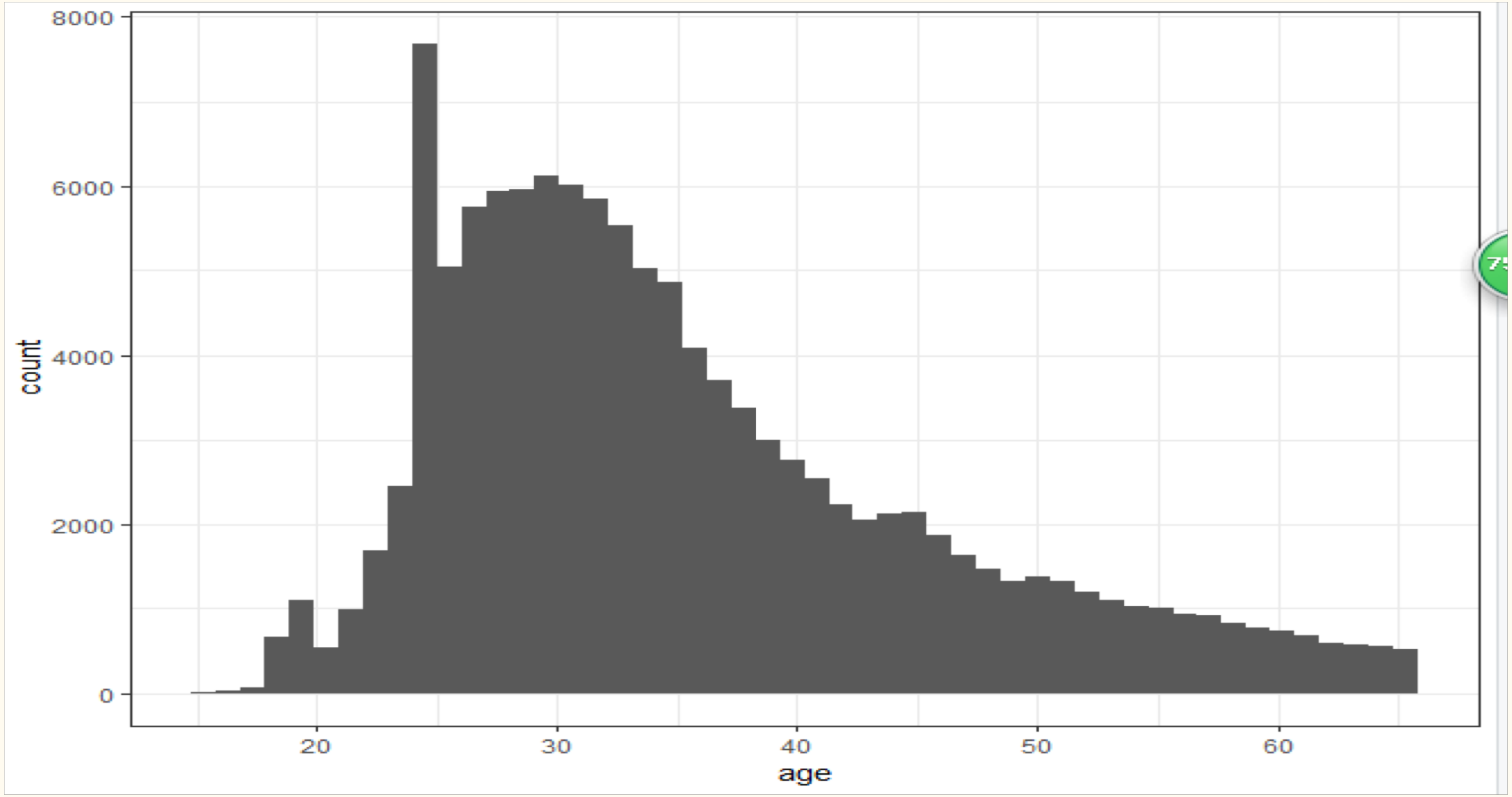
Data Visualization and Analysis



Data Visualization and Analysis



Data Visualization and Analysis



Data Analysis for 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 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

Data Preprocessing.

```r
#remove date_first_booking
df = df[-c(which(colnames(df) %in% c('date_first_booking')))]
# replace missing values
for(i in 1:ncol(df)){
  df[is.na(df[,i]), i] <- mean(df[,i], na.rm = TRUE)
}
```

```r

# split date_account_created in year, month and day
dac = as.data.frame(str_split_fixed(df$date_account_created, '-',
df['dac_year'] = dac[,1]
df['dac_month'] = dac[,2]
df['dac_day'] = dac[,3]
df = df[,-c(which(colnames(df) %in% c('date_account_created')))]
```

Data Preprocessing

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’.

Data Preprocessing

ONE HOT ENCODING

country_destination
NDF
NDF
US
other
US
US
US
US
US
US
US
NDF
FR
NDF
NDF
CA

AU	CA	DE	ES	FR	GB	IT	NDF	NL	Other	PT	US
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0

Data Preprocessing

ONE HOT ENCODING

```
# 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)
```

Models used for Analysis

NAIVE BAYES
KNN
ARTIFICIAL NEURAL NETWORKS (ANN)
RANDOM FOREST
C50
MULTI-CLASSIFICATION

NAIVE BAYES

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

```
      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
>
```

KNN

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

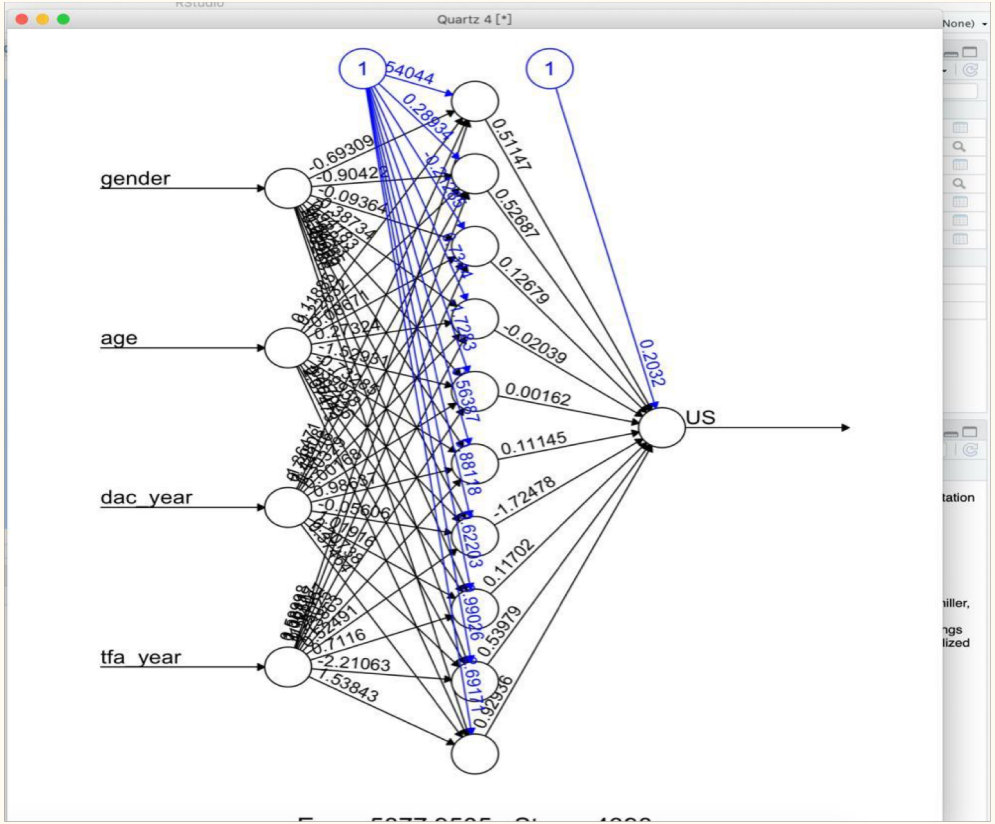
```
kknn    0    1
   0 8923 3069
   1 1266  973
> knn_error_rate=sum(fit!=test$US)/length(test$US)
> print(knn_error_rate)
[1] 0.3046167
> accuracy <- (1-knn_error_rate)*100
> accuracy
[1] 69.53833
>
```

ANN

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

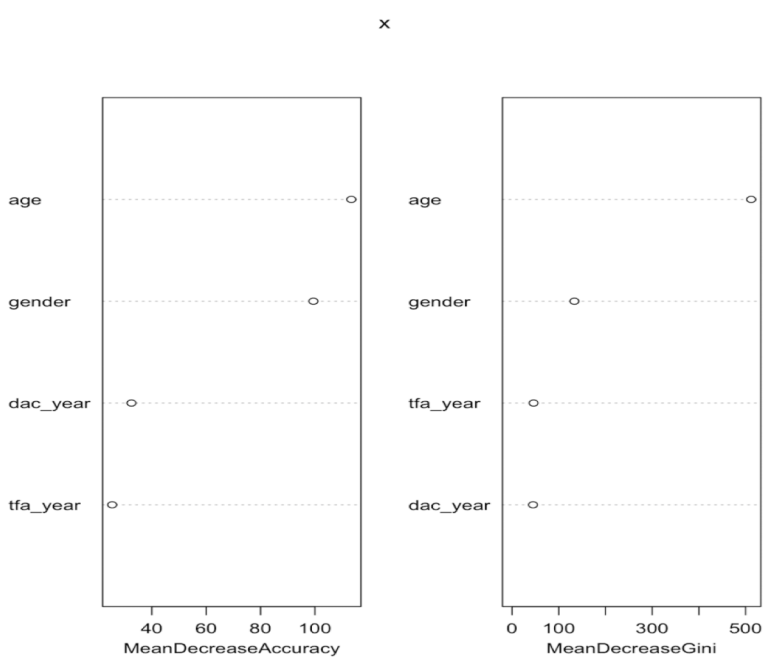
ANN

```
35:1  (Top Level)

Console  Terminal

~/

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



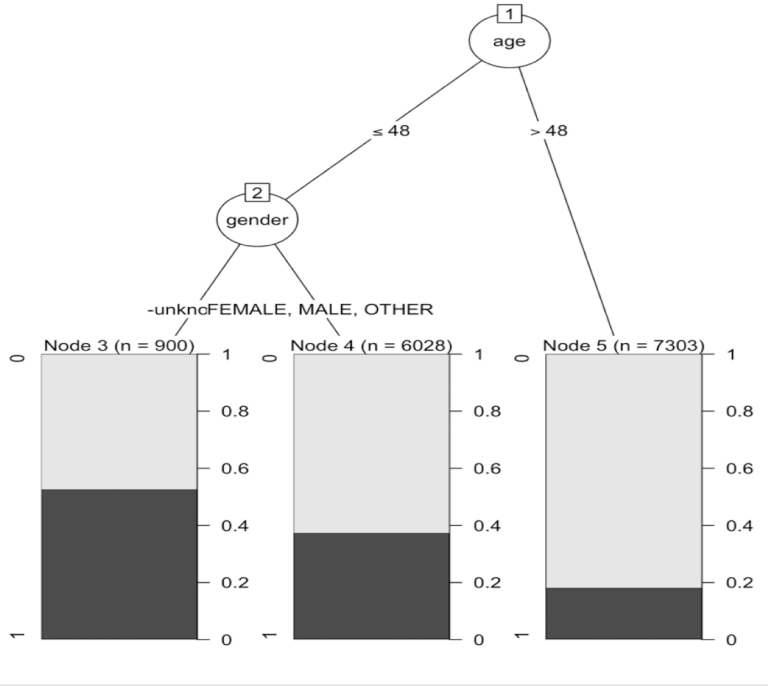
RANDOM FOREST

```
> library(randomForest)
> Airbnb <- read.csv("~/Desktop/Airbnb/nb/df_final.csv")
> set.seed(123)
> index <-seq(1, nrow(Airbnb), by = 3)
> Airbnb <- Airbnb[index,]
> index <-seq(1, nrow(Airbnb), by = 5)
> test <- Airbnb[index,]
> train <- Airbnb[-index,]
> train<- train[c(2,3,13,16,19)]
> test<-test[c(2,3,13,16,19)]
> x<-randomForest(factor(US)~.,data=test, importance=TRUE, ntree=1000)
> importance(x)
                0          1 MeanDecreaseAccuracy MeanDecreaseGini
gender    111.95694 -104.43637             99.43009         133.40220
age       111.88230  -25.74519            113.39732         511.79970
dac_year   34.48575  -33.16318             32.50266          44.94029
tfa_year   31.09099  -31.43054             25.41277          46.13287
> varImpPlot(x)
> Prediction<-predict(x,test)
> table(actual=test[,5],Prediction)
      Prediction
actual    0    1
     0 9884  305
     1 3539  503
> wrong<-(test[,5]!=Prediction)
> error_rate<-sum(wrong)/length(wrong)
> error_rate
[1] 0.2701145
> accuracy <- (1-error_rate)*100
> accuracy
[1] 72.98855
>
```



C50

```
age > 48: 0 (7303/1319)
age <= 48:
:...gender = -unknown-: 1 (900/426)
    gender in {FEMALE,MALE,OTHER}: 0 (6028/2249)


Evaluation on training data (14231 cases):

           Decision Tree
         ----------------
         Size      Errors

            3 3994(28.1%)   <<


          (a)   (b)    <-classified as
         ----  ----
         9763   426    (a): class 0
         3568   474    (b): class 1


        Attribute usage:

        100.00% age
         48.68% gender


Time: 0.0 secs

> plot(C50_class)
> C50_predict<-predict( C50_class ,test , type="class" )
> table(actual=test[,5],C50=C50_predict)
      C50
actual    0    1
     0 9763  426
     1 3568  474
> wrong<- (test[,5]!=C50_predict)
> c50_rate<-sum(wrong)/length(test[,4])
> c50_rate
[1] 0.2806549
> accuracy <- (1-c50_rate)*100
> accuracy
[1] 71.93451
>
```

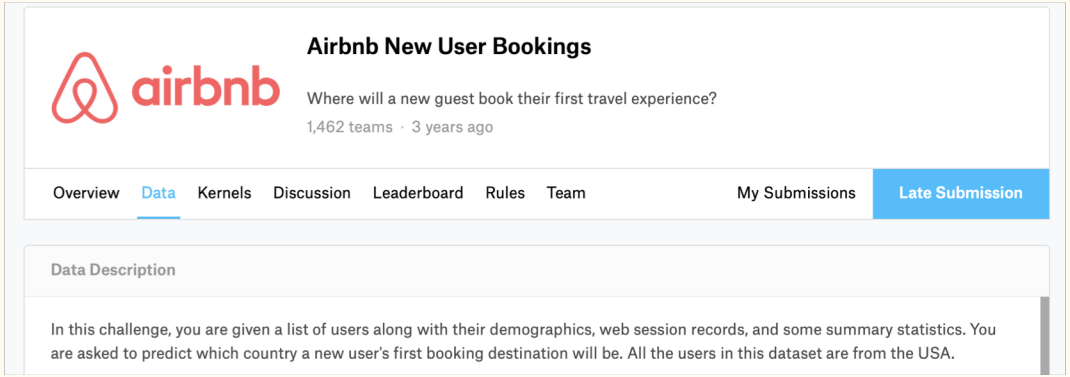


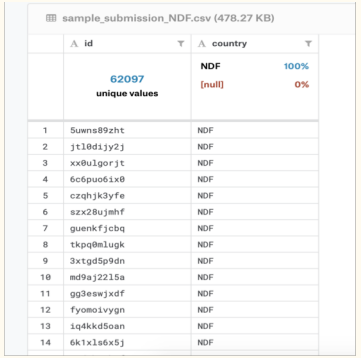
Comparing accuracies of different models

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

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...)