R Notebook

Cover Page

Data Mining and Predictive Analysis :

Project Title: Exploratory and Predictive Analysis of Airbnb Austin Market for finding high booking properties

Market Assigned to the team: Austin “We, the undersigned, certify that the report submitted is our own original work, all authors participated in work in a substantive way; all authors have seen and approved the report as submitted, the text, images, illustrations, and other items included in the manuscript do not carry any infringement/ plagiarism issue upon any existing copyrighted materials.”

Signature Names of the signed team members

Dharkar Gauri

Ishwar Dodamani Vishal

Jethwa Shlok

Joshi Jay

Parekh Malav

Shah Ami

Executive Summary

The project is focused on finding insights about the Airbnb Austin market for a property to become profitable to an investor. Austin is a developing market and has a lot of potential in real estate. Due to the presence of educational hubs, tourist spots, different festivals, and a growing job market, people ar e attracted to visit Austin. Effective analysis of this market will prove ben eficial for investors as well as customers.

Properties in Austin are comparatively less expensive than other big cities. This implies that taking into consideration a few important aspects of the pr operty, an investor can gain profit in no time. The data makes generalized as sumptions about the majority of people. As the analysis is based on what the majority of customers prefer, investors are recommended to provide that. This analysis thereby fulfills customer expectations as well. After analyzing the Airbnb data for Austin, we came up with meaningful insights for a buyer. Inve stors should buy an apartment, guest house, condominium, or a guest suite whi ch has year wide high booking rates. Customers mainly prefer the properties

```
having lower prices. Also, review scores are an important consideration conce
rning customers. Review scores are directly proportional to the booking rate.

From the investors' perspective, our emphasis was not to miss high booking pr
operties. All the models are developed with this premise. Different models en
counter different business problems. The first business case suggests focusin
g on year wide availability and amenities to attract more customers. Secondly
, super host status, neighborhood, different fees (cleaning, security),  and
review scores matter utmost in price predictions and property management. Fin
ally, people's preference for scenic properties and different amenities affec
ts the upgrading and renovation aspect of the estate.
```

Research Questions

What is the best time to visit Austin? Spring and fall are the best times to visit Austin. From September to November and from March to May, the temperatures are perfect for a visit to Austin. Austin is host to ACL festival and SXSW which also takes place in these months and which are most popular among people.

What should investors focus on before purchasing the property? Investors should focus on the popularity of the area where the house resides i.e to find which areas are mostly visited by the customers. What should be the availability of the house? Select a house which could be available more often (In the 30 day or 365-day window.)

What amenities do people expect in an Airbnb house? Bathrooms, bedrooms, Leisure, Safety, Health, Kitchen are some amenities that people look for before booking an Airbnb house.

Which are the most popular property types which people look for on the Airbnb Austin market? GuestHouse, GuestSuite, Condominium, Apartment are some of the most booked properties in Austin.

Are review scores related to the property bookings? Yes, they are. Normally review scores of 9 or 10 are only preferred by other people before they book their property.

What factors must be considered before buying a property? The locality of the property, availability of the property, the number of people it accommodates, the number of bathrooms, and bedrooms must be considered.

What factors must be considered for deciding the price of a property? * Pricing mostly depends on the availability period, cleaning fee, accommodation, amenities whether the host is a super host or not. * How will the owner manage the property after the acquisition of the property? * Better review scores, host response time, host response rate help improve management of the property. * Will upgrades and renovations help in increasing the booking rate? * People usually prefer Scenic views and leisure activities. Upgrading to these amenities can be really worthwhile.

```
austinMarketTrain<-read_csv('austinMarketTrainClean_v1.0.csv')
```

```
austinMarketTrain2<-read.csv('austinMarketTrainClean_v1.0.csv')
dfAustinClean <- read.csv('Train.csv')
```

Modeling

Data Preparation

We started off with cleaning our data, which started with by replacing empt spaces by NA, removing unwanted characters, performing data type conversion, clubbing of some variables to reduce the number of levels it had and to make more sense out of it and lastly making dummy variables for the desired variables.

For data preparation:

- We clubbed the zipcodes into 7 groups depending on their geographical location and other research of the neighborhood that we did.
- Next, we created 11 dummy variables for amenities based on their significance and relevance to the Texan market.
- We also divided host_listing_count into 4 categories. The reason behind that was to check if the professional management of these acquired properties affects the high booking rate.

Replacing empty spaces by NA

```
dfAustinClean <-
dfAustinClean[rowSums(is.na(dfAustinClean))!=ncol(dfAustinClean), ] # first s
et blank to NA
```

Removing $ from cleaning_fee, security_deposit,weekly_price, monthly_price, extra_people and price and changing to numeric

```
dfAustinClean$cleaning_fee <- stringr::str_replace(dfAustinClean$cleaning_fee
, '\\$', '')
dfAustinClean$cleaning_fee <- as.numeric(dfAustinClean$cleaning_fee)

dfAustinClean$price <- stringr::str_replace(dfAustinClean$price, '\\$', '')
dfAustinClean$price <- stringr::str_replace(dfAustinClean$price, ',', '')
dfAustinClean$price <- as.numeric(dfAustinClean$price)


dfAustinClean$extra_people <- stringr::str_replace(dfAustinClean$extra_people
, '\\$', '')
dfAustinClean$extra_people <- as.numeric(dfAustinClean$extra_people)

dfAustinClean$monthly_price <- stringr::str_replace(dfAustinClean$monthly_pri
ce, '\\$', '')
dfAustinClean$monthly_price <- as.numeric(dfAustinClean$monthly_price)
```

```
dfAustinClean$security_deposit <- stringr::str_replace(dfAustinClean$security
_deposit, '\\$', '')
dfAustinClean$security_deposit <- as.numeric(dfAustinClean$security_deposit)

dfAustinClean$weekly_price <- stringr::str_replace(dfAustinClean$weekly_price
, '\\$', '')
dfAustinClean$weekly_price <- as.numeric(dfAustinClean$weekly_price)
```

Removing % from host_reponse_rate and changing to numeric

```
dfAustinClean$host_response_rate <- stringr::str_replace(dfAustinClean$host_r
esponse_rate, '\\%', '')
dfAustinClean$host_response_rate <- as.numeric(dfAustinClean$host_response_ra
te)
```

Removing redundant columns state and market from dataframe

```
dfAustinClean$state <- NULL
dfAustinClean$market <- NULL

austinMarketTrain1 <- dfAustinClean
```

replacing the NA in host_response_rate with 0

```
austinMarketTrain1$host_response_rate<-austinMarketTrain1$host_response_rate
%>%
  replace_na(0)
```

replacing the NA in zipcode with 0

```
austinMarketTrain1$zipcode<-austinMarketTrain1$zipcode %>%
  replace_na(0)
```

Grouping zipcodes based on their locality

```
Group1 <- c(78652,78701,78703,78705,78712)
Group2 <- c(78719,78721,78722,78723)
Group3 <- c(78717,78724,78725,78726,78727,78728,78729)
Group4 <- c(78730,78731,78732,78733,78734,78738,78739)
Group5 <- c(78735,78736,78737,78741,78749)
Group6 <- c(78742,78744,78745,78747,78748,78750,78751,78752,78753,78754,78756
,78757,78758,78759)
Misc <- c(78613,78620,78650,78653,78660,78669,78746,78767,80211,78712)

austinMarketTrain1$category <- ifelse(austinMarketTrain1$zipcode %in% Group1,
"Group 1",
                                      ifelse(austinMarketTrain1$zipcode %in% Group2
, "Group 2",
                                      ifelse(austinMarketTrain1$zipcode %in% Group3
,"Group 3",
                                      ifelse(austinMarketTrain1$zipcode %in% Group4
,"Group 4",
                                      ifelse(austinMarketTrain1$zipcode %in% Group5
```

```
,"Group 5",
                                    ifelse(austinMarketTrain1$zipcode %in% Group6
,"Group 6",
                                           "Misc"))))))
```

Grouping the Host_listings_count based on different ranges

```
Local <- seq(0, 1, by=1)
Small <- c(2,3,4)
Medium <- seq(5,25, by=1)
Large <- seq(26,1000, by=1)
Misc1<- c(1717, 1780,1826)

austinMarketTrain1$host_listings_count <- ifelse(austinMarketTrain1$host_list
ings_count %in% Local, "Local",
                                    ifelse(austinMarketTrain1$host_listings_count
%in% Small, "Small",
                                    ifelse(austinMarketTrain1$host_listings_count
%in% Medium,"Medium",
                                    ifelse(austinMarketTrain1$host_listings_count
%in% Large,"Large",
                                    ifelse(austinMarketTrain1$host_listings_count
%in% Misc1,"Miscellanious","Other")))))

Host_listin_Time_levels = c('within an hour',' within a day','within a few ho
urs','a few days or more')

#austinMarketTrain1$host_response_time <- ifelse(austinMarketTrain1$host_resp
onse_time %in% Host_listin_Time_levels, austinMarketTrain1$host_response_time
,-1)
```

Making dummies for host_listings_count and host_response_time

```
library("fastDummies")
austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "host_listings_count")

austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "host_response_time")
```

Replacing NAs in host_response_rate with 0

```
austinMarketTrain1$host_response_rate<-austinMarketTrain1$host_response_rate
%>%
  replace_na(0)
```

Categorizing the host_response rate based on range of rates mentioned below: *Weak: 0-50%* Average: 51-75% *Abover Average: 76-85%* Marvelous: above 95%

```
weak <- seq(0, 50, by=1)
average <- seq(51, 75, by=1)
```

```
aboveAverage <- seq(76,95, by=1)
marvelous<- seq(96,100, by=1)

austinMarketTrain1$host_response_rate <- ifelse(austinMarketTrain1$host_respo
nse_rate %in% weak, "Weak",
                                    ifelse(austinMarketTrain1$host_response_rate
%in% average, "Average",
                                    ifelse(austinMarketTrain1$host_response_rate
%in% aboveAverage,"Above Average",
                                    ifelse(austinMarketTrain1$host_response_rate
%in% marvelous,"Marvelous","Other"))))
```

Making dummies for host_response_rate, zipcodes category, bed type and room types

```
austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "host_response_rate")
austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "category")
austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "bed_type")
austinMarketTrain1 <- fastDummies::dummy_cols(austinMarketTrain1, select_colu
mns = "room_type")

austinMarketTrain2<-austinMarketTrain1
```

Repalcing NA values in the numeric columns with mode

```
val <- unique(austinMarketTrain2$beds[!is.na(austinMarketTrain2$beds)])
# Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$beds, val)))] # Mode
of vec_miss



austinMarketTrain2$beds <- austinMarketTrain2$beds
# Replicate vec_miss
austinMarketTrain2$beds[is.na(austinMarketTrain2$beds)] <- mode



val <- unique(austinMarketTrain2$bedrooms[!is.na(austinMarketTrain2$bedrooms)
])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$bedrooms, val)))] # M
ode of vec_miss



austinMarketTrain2$bedrooms <- austinMarketTrain2$bedrooms
# Replicate vec_miss
austinMarketTrain2$bedrooms[is.na(austinMarketTrain2$bedrooms)] <- mode

val <- unique(austinMarketTrain2$bathrooms[!is.na(austinMarketTrain2$bathroom
s)])              # Values in vec_miss
```

```
mode <- val[which.max(tabulate(match(austinMarketTrain2$bathrooms, val)))] #
Mode of vec_miss


austinMarketTrain2$bathrooms <- austinMarketTrain2$bathrooms
# Replicate vec_miss
austinMarketTrain2$bathrooms[is.na(austinMarketTrain2$bathrooms)] <- mode

val <- unique(austinMarketTrain2$cleaning_fee[!is.na(austinMarketTrain2$clean
ing_fee)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$cleaning_fee, val)))]
# Mode of vec_miss


austinMarketTrain2$cleaning_fee <- austinMarketTrain2$cleaning_fee
# Replicate vec_miss
austinMarketTrain2$cleaning_fee[is.na(austinMarketTrain2$cleaning_fee)] <- mo
de

val <- unique(austinMarketTrain2$weekly_price[!is.na(austinMarketTrain2$weekl
y_price)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$weekly_price, val)))]
# Mode of vec_miss


austinMarketTrain2$weekly_price <- austinMarketTrain2$weekly_price
# Replicate vec_miss
austinMarketTrain2$weekly_price[is.na(austinMarketTrain2$weekly_price)] <- mo
de

val <- unique(austinMarketTrain2$security_deposit[!is.na(austinMarketTrain2$s
ecurity_deposit)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$security_deposit, val
)))] # Mode of vec_miss


austinMarketTrain2$security_deposit <- austinMarketTrain2$security_deposit
# Replicate vec_miss
austinMarketTrain2$security_deposit[is.na(austinMarketTrain2$security_deposit
)] <- mode

val <- unique(austinMarketTrain2$review_scores_cleanliness[!is.na(austinMarke
tTrain2$review_scores_cleanliness)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_cleanli
ness, val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_cleanliness <- austinMarketTrain2$review_sco
res_cleanliness                                   # Replicate vec_miss
```

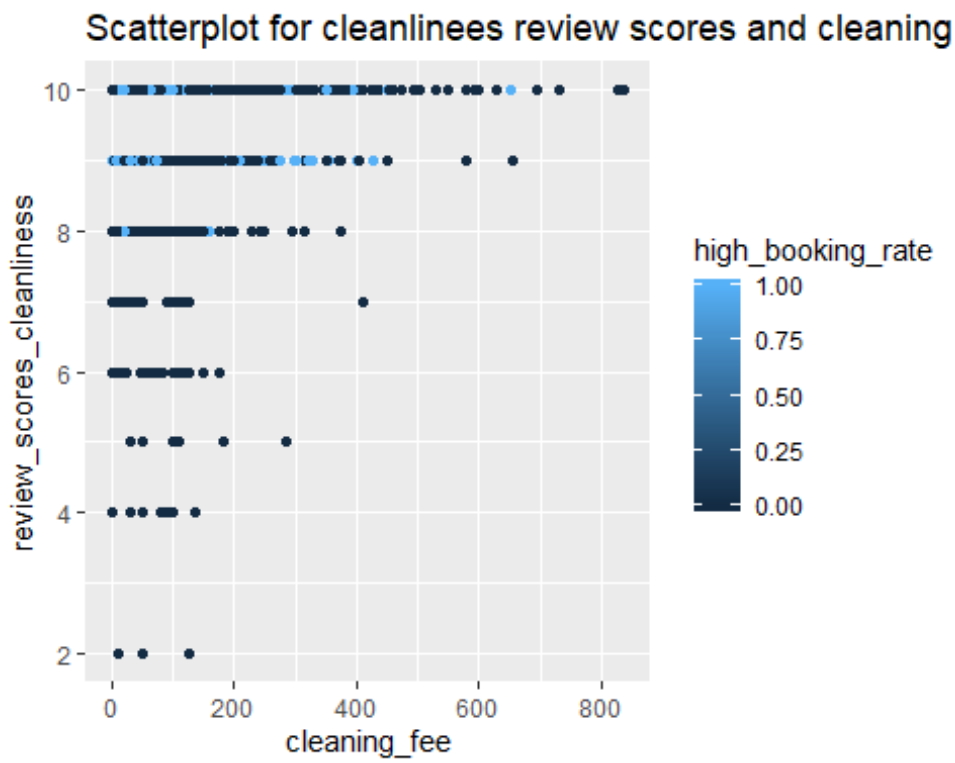
```
austinMarketTrain2$review_scores_cleanliness[is.na(austinMarketTrain2$review_
scores_cleanliness)] <- mode

val <- unique(austinMarketTrain2$review_scores_checkin[!is.na(austinMarketTra
in2$review_scores_checkin)])             # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_checkin
, val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_checkin <- austinMarketTrain2$review_scores_
checkin                                   # Replicate vec_miss
austinMarketTrain2$review_scores_checkin[is.na(austinMarketTrain2$review_scor
es_checkin)] <- mode

val <- unique(austinMarketTrain2$review_scores_communication[!is.na(austinMar
ketTrain2$review_scores_communication)])             # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_communi
cation, val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_communication <- austinMarketTrain2$review_s
cores_communication                                  # Replicate vec_miss
austinMarketTrain2$review_scores_communication[is.na(austinMarketTrain2$revie
w_scores_communication)] <- mode

val <- unique(austinMarketTrain2$review_scores_location[!is.na(austinMarketTr
ain2$review_scores_location)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_locatio
n, val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_location <- austinMarketTrain2$review_scores
_location                                    # Replicate vec_miss
austinMarketTrain2$review_scores_location[is.na(austinMarketTrain2$review_sco
res_location)] <- mode

val <- unique(austinMarketTrain2$review_scores_rating[!is.na(austinMarketTrai
n2$review_scores_rating)])             # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_rating,
val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_rating <- austinMarketTrain2$review_scores_r
ating                                    # Replicate vec_miss
austinMarketTrain2$review_scores_rating[is.na(austinMarketTrain2$review_score
s_rating)] <- mode

val <- unique(austinMarketTrain2$review_scores_value[!is.na(austinMarketTrain
2$review_scores_value)])             # Values in vec_miss
```

```
mode <- val[which.max(tabulate(match(austinMarketTrain2$review_scores_value,
val)))] # Mode of vec_miss


austinMarketTrain2$review_scores_value <- austinMarketTrain2$review_scores_va
lue                                    # Replicate vec_miss
austinMarketTrain2$review_scores_value[is.na(austinMarketTrain2$review_scores
_value)] <- mode

val <- unique(austinMarketTrain2$host_listings_count[!is.na(austinMarketTrain
2$host_listings_count)])              # Values in vec_miss
mode <- val[which.max(tabulate(match(austinMarketTrain2$host_listings_count,
val)))] # Mode of vec_miss


austinMarketTrain2$host_listings_count <- austinMarketTrain2$host_listings_co
unt                                    # Replicate vec_miss
austinMarketTrain2$host_listings_count[is.na(austinMarketTrain2$host_listings
_count)] <- mode

austinMarketTrain1 <- austinMarketTrain2
```

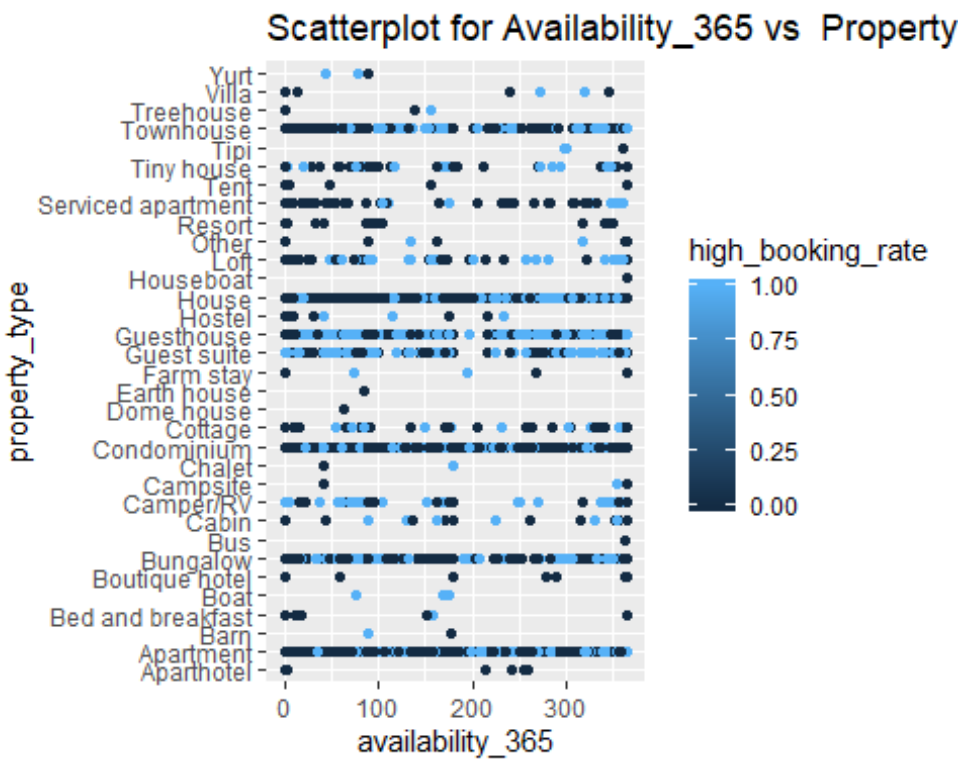
Exploratory Data Analysis:

```
austinMarketEDA <- read_csv('Train.csv')

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   access = col_character(),
##   bed_type = col_character(),
##   cancellation_policy = col_character(),
##   city = col_character(),
##   cleaning_fee = col_character(),
##   description = col_character(),
##   extra_people = col_character(),
##   host_about = col_character(),
##   host_acceptance_rate = col_logical(),
##   host_has_profile_pic = col_logical(),
##   host_identity_verified = col_logical(),
##   host_is_superhost = col_logical(),
##   host_location = col_character(),
##   host_neighbourhood = col_character(),
##   host_response_rate = col_character(),
##   host_response_time = col_character(),
##   host_since = col_date(format = ""),
##   host_verifications = col_character(),
##   house_rules = col_character(),
##   instant_bookable = col_logical()
```

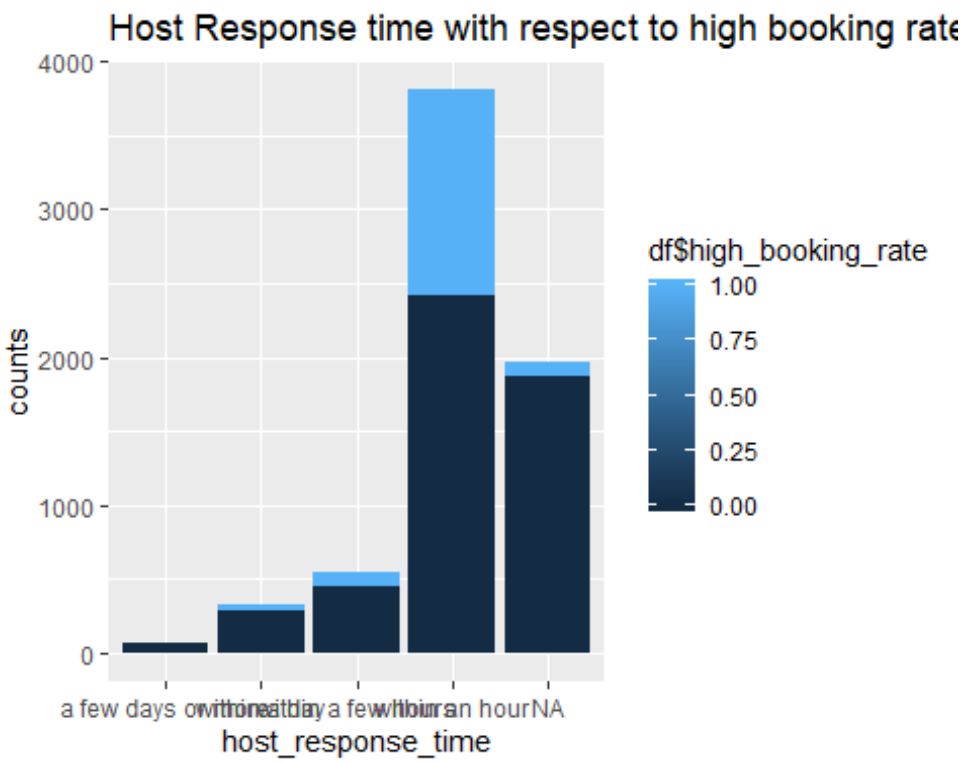
```
##   # ... with 19 more columns
## )

## See spec(...) for full column specifications.

austinMarketEDA1 <- read_csv('AustinMarketTrain.csv')

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   access = col_character(),
##   bed_type = col_character(),
##   cancellation_policy = col_character(),
##   city = col_character(),
##   description = col_character(),
##   host_about = col_character(),
##   host_acceptance_rate = col_logical(),
##   host_has_profile_pic = col_logical(),
##   host_identity_verified = col_logical(),
##   host_is_superhost = col_logical(),
##   host_listings_count = col_character(),
##   host_location = col_character(),
##   host_neighbourhood = col_character(),
##   host_response_rate = col_character(),
##   host_response_time = col_character(),
##   host_since = col_date(format = ""),
##   host_verifications = col_character(),
##   house_rules = col_character(),
##   instant_bookable = col_logical(),
##   interaction = col_character()
##   # ... with 13 more columns
## )
## See spec(...) for full column specifications.

austinMarketEDA1 %>%
ggplot( aes(y = review_scores_cleanliness, x = cleaning_fee, color = high_boo
king_rate)) + geom_point() + ggtitle("Scatterplot for cleanlinees review scor
es and cleaning fee")
```



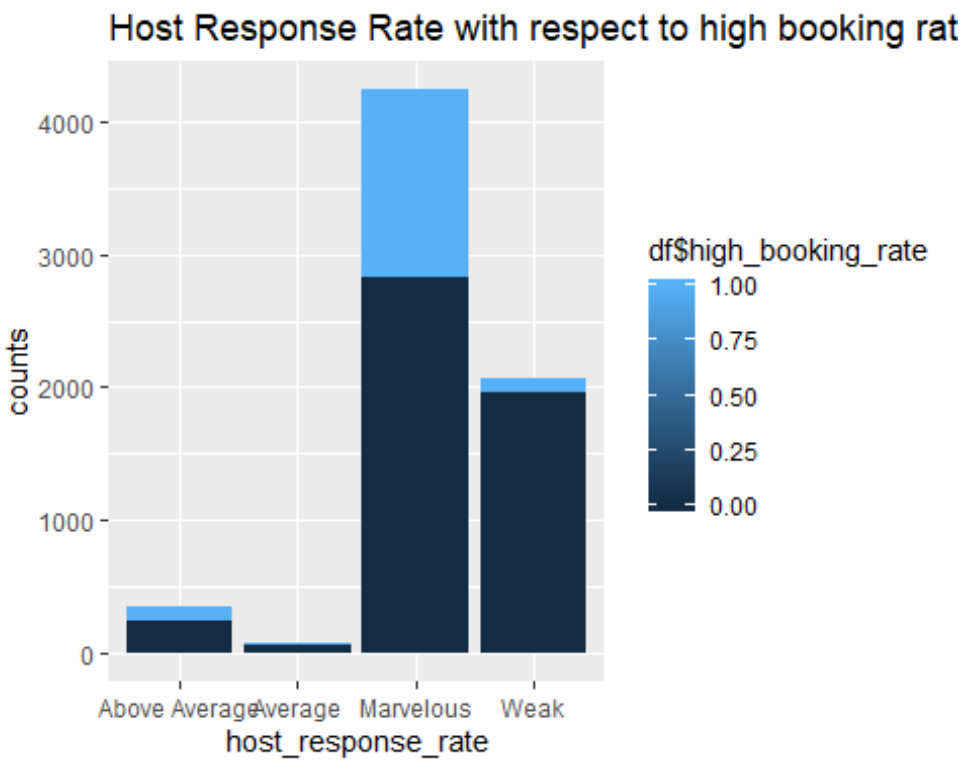

```
austinMarketEDA1 %>%
ggplot( aes(y = property_type, x = availability_365, color = high_booking_rat
e)) + geom_point() + ggtitle("Scatterplot for Availability_365 vs  Property T
ype")
```



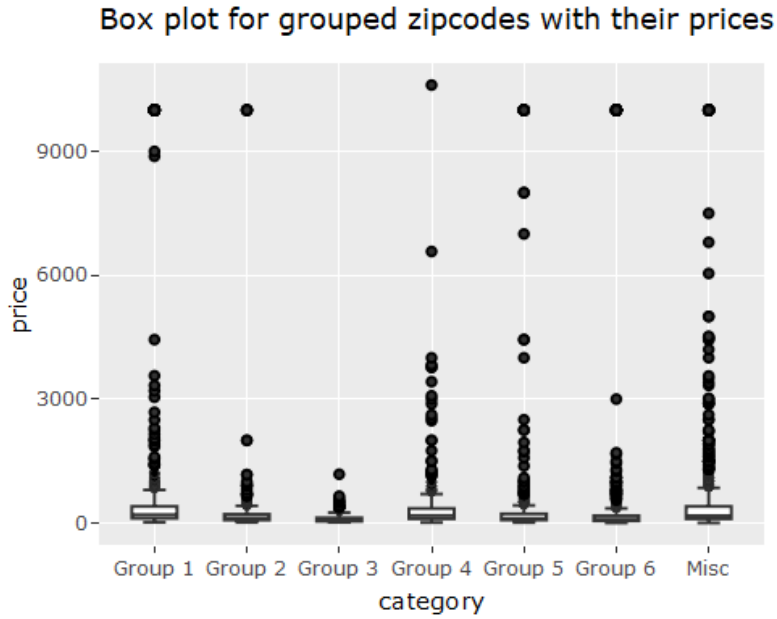

```
df <- austinMarketEDA %>%
  #filter(high_booking_rate == 0)%>%
  group_by(high_booking_rate,host_response_time) %>%
  summarise(counts = n())

ggplot(df, aes(x = host_response_time, y = counts, fill = df$high_booking_rat
e)) + ggtitle("Host Response time with respect to high booking rates")+
  geom_histogram( stat = "identity")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
df <- austinMarketEDA1 %>%
  #filter(high_booking_rate == 0)%>%
  group_by(high_booking_rate,host_response_rate) %>%
  summarise(counts = n())


ggplot(df, aes(x = host_response_rate, y = counts, fill = df$high_booking_rat
e)) + ggtitle("Host Response Rate with respect to high booking rates")+
  geom_histogram( stat = "identity")

## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
highplot<- austinMarketEDA1%>%
  #filter(high_booking_rate == 1)%>%
  ggplot( aes(x=category, y = price))  + geom_boxplot() + ggtitle("Box plot f
or grouped zipcodes with their prices")
ggplotly(highplot)
```



Method and research:

We figured the Texan market would be a bit different as compared to all the other markets. So, we did immense research and found out that there were some specific things that were particularly relevant. We have tried to factor in as many variables as we could as there was extensive cleaning.

The main focus was on selecting these cleaned variables on the models and **increasing the sensitivity.** The reason behind that is three-fold.

- Firstly, houses in Texas are relatively cheaper as compared to the other parts of the country.
- Secondly, we thought that a low sensitivity would lead to a larger number of options for the investors.
- Thirdly, we believe that some of the properties may not have a high booking rate given their current condition. Using our study, we are suggesting renovations and upgrades that could be made to property. So, if that is done, the investor has a high chance at getting profitable

Having said that, we took on another important task - choosing an appropriate method for modelling. We again turned to market research for that. We found out that all the data was divided into many classes and categories and at the same time. Hence, we thought it would be best to use a bagged decision tree or KNN. And that's why we used the bagged decision tree and KNN for our several models and chose based on sensitivity.

Reason for KNN : KNN best capturers local behavior. As the data was concentrated in one city(Austin), we chose to use KNN. In addition to that, it works well with large datasets and is a non-parametric method. This implied that minimum initializations and computations were to be made from our end to achieve high performance measures. [Refer appendix 4]

Reason for bagged trees: Bagged trees helps to reduce variance of decision trees. Since our problem was for classification, bagged trees perform well. Also, our data was large and ensemble methods are known to deal with large data sets. Bagged trees perform well with varying the number of base estimators and the size of base estimators. Outputs from bagged trees can help identify subsets of input variables that may be most or least relevant to the problem and suggest a possible feature selection experiment that we can perform to achieve higher accuracy. [Refer appendix 3]

```
library("caret")
#austinMarketTrain1$high_booking_rate<-as.factor(austinMarketTrain1$high_book
ing_rate)
set.seed(123)
austinCleanData <- read_csv('AustinMarketTrain.csv')

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   access = col_character(),
##   bed_type = col_character(),
##   cancellation_policy = col_character(),
##   city = col_character(),
##   description = col_character(),
##   host_about = col_character(),
##   host_acceptance_rate = col_logical(),
##   host_has_profile_pic = col_logical(),
##   host_identity_verified = col_logical(),
##   host_is_superhost = col_logical(),
##   host_listings_count = col_character(),
##   host_location = col_character(),
##   host_neighbourhood = col_character(),
##   host_response_rate = col_character(),
##   host_response_time = col_character(),
##   host_since = col_date(format = ""),
##   host_verifications = col_character(),
##   house_rules = col_character(),
##   instant_bookable = col_logical(),
##   interaction = col_character()
##   # ... with 13 more columns
## )

## See spec(...) for full column specifications.
```

```
austinMarketTrainFinal <- austinMarketTrain1 %>% sample_frac(0.7)
austinMarketTestFinal <- dplyr::setdiff(austinMarketTrain1, austinMarketTrain
Final)

df<-read.csv('AustinMarketTrain.csv')
df1<-df
df1$high_booking_rate<-as.factor(df1$high_booking_rate)

set.seed(123)
austinTrainSet <- df %>% sample_frac(0.70)
austinTestSet <- dplyr::setdiff(df, austinTrainSet)

set.seed(123)
austinTrainSet_f <- df1 %>% sample_frac(0.70)
austinTestSet_f <- dplyr::setdiff(df1, austinTrainSet_f)
```

Initial Acquisition model

```
fitBaggedTree <- train(high_booking_rate ~ is_location_exact+category+accommo
dates+availability_30+availability_365 + bedrooms + bathrooms, data= austinTr
ainSet_f, method='treebag', trControl=trainControl(method='cv', number=10))
#The default bootstrap replications is 25. You can change it using the argume
nt nbagg = XX
```

Management Model

```
fitBaggedTree1 <- train(high_booking_rate ~ review_scores_checkin+review_scor
es_cleanliness+review_scores_communication+review_scores_location+review_scor
es_rating+review_scores_value+host_is_superhost+host_response_rate+cancellati
on_policy+instant_bookable+security_deposit+host_listings_count+host_response
_time+host_identity_verified+Bathroom+Leisure+Other+health, data= austinTrain
Set_f, method='treebag', trControl=trainControl(method='cv', number=10))
#The default bootstrap replications is 25. You can change it using the argume
nt nbagg = XX
```

Pricing Model

```
fitBaggedTree2 <- train(high_booking_rate ~accommodates+bathrooms+bedrooms+av
ailability_90+availability_365+cleaning_fee+host_is_superhost+room_type+categ
ory+host_listings_count+Bathroom+Leisure+Other+Scenic+health, data= austinTra
inSet_f, method='treebag', trControl=trainControl(method='cv', number=10))
#The default bootstrap replications is 25. You can change it using the argume
nt nbagg = XX
```

Upgrades and Renovations

```
fitKNN <- train(as.factor(high_booking_rate) ~  bathrooms + bedrooms + beds
+ instant_bookable  + room_type  + Bathroom + Kitchen + Leisure + Other  + Sa
fety + Scenic,data=austinTrainSet_f, trControl=trainControl(method='cv', numb
er=100), preProcess = c("center", "scale"), tuneLength = 3,method='knn')
```

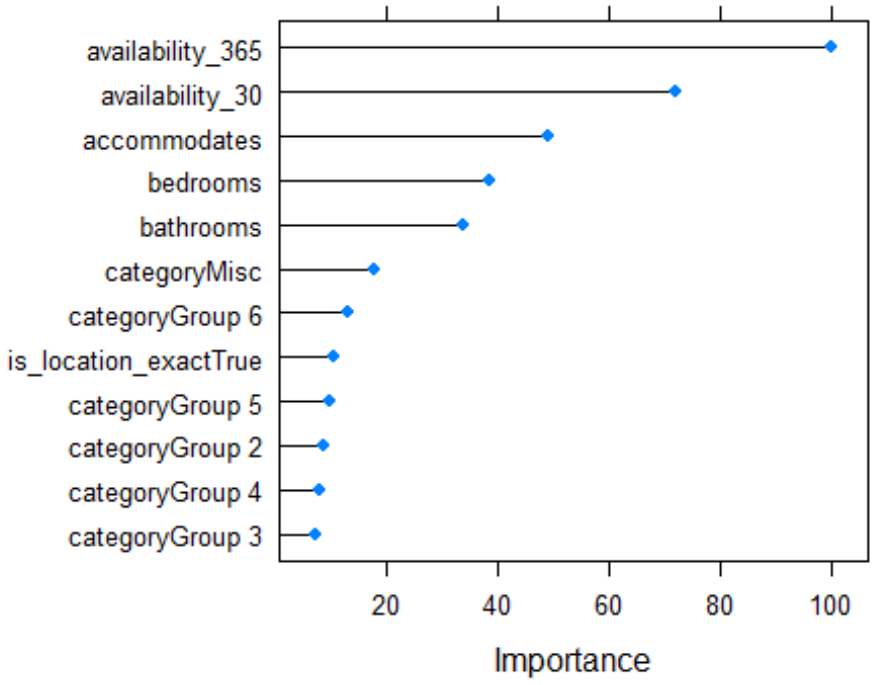
Results and Findings

We were able to answer most of the questions from the deliverable 1. Questions included which areas should the investor invest in? what all should one keep in mind while investing? costliest areas in Austin, and the factors leading to a high booking rate. All of these were accomplished by applying models to different business cases - initial acquisition of homes, pricing and management, and upgrades to a home in Austin. All of these business cases had different factors which were important and we can see the factors which were important in each business case as follows:

Business Cases

Initial Acquisition:

```
#See the variables plotted by importance (according to the bagged tree):
plot(varImp(fitBaggedTree), top= 12)
```

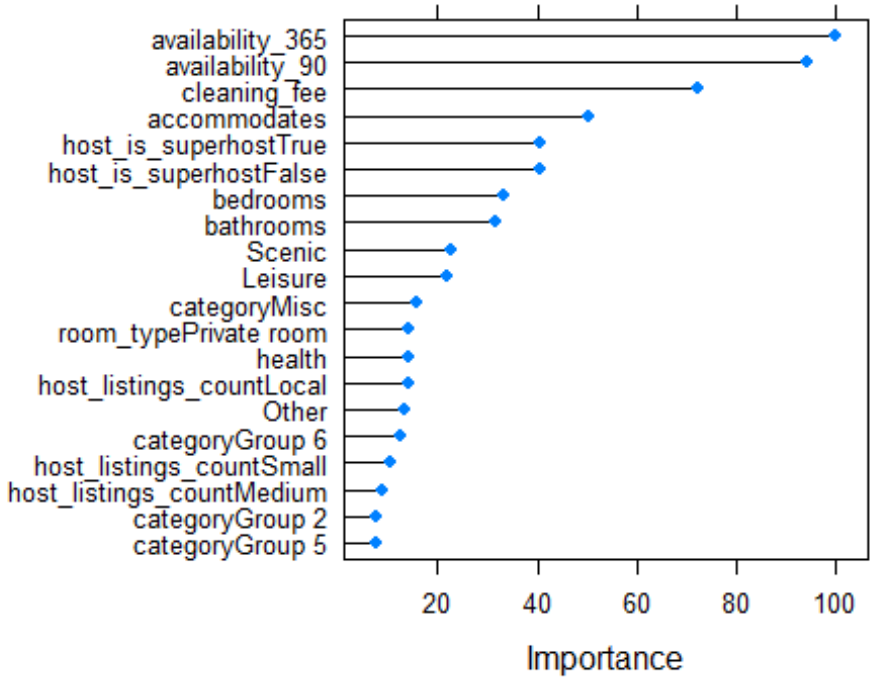



```
#Make predictions:
resultsBaggedTree1 <-
  fitBaggedTree %>%
  predict(austinTestSet_f, type='prob') %>%
  bind_cols(austinTestSet_f, predictedProb=.)
resultsBaggedTree1<- resultsBaggedTree1%>%
  mutate(predictedClass = as.factor(ifelse(resultsBaggedTree1$`1` > 0.35, 1,
```

```
0)))

resultsBaggedTree1 %>%
  xtabs(~predictedClass+ high_booking_rate, .) %>%
  confusionMatrix(positive = '1') #%>%

## Confusion Matrix and Statistics
##
##                high_booking_rate
## predictedClass    0    1
##              0 1172  199
##              1  368  281
##
##                Accuracy : 0.7193
##                  95% CI : (0.6992, 0.7388)
##     No Information Rate : 0.7624
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.309
##
##  Mcnemar's Test P-Value : 1.722e-12
##
##             Sensitivity : 0.5854
##             Specificity : 0.7610
##          Pos Pred Value : 0.4330
##          Neg Pred Value : 0.8549
##              Prevalence : 0.2376
##          Detection Rate : 0.1391
##    Detection Prevalence : 0.3213
##       Balanced Accuracy : 0.6732
##
##        'Positive' Class : 1
##
```

Pricing Model:

```
#See the variables plotted by importance (according to the bagged tree):
plot(varImp(fitBaggedTree2), top=20)
```

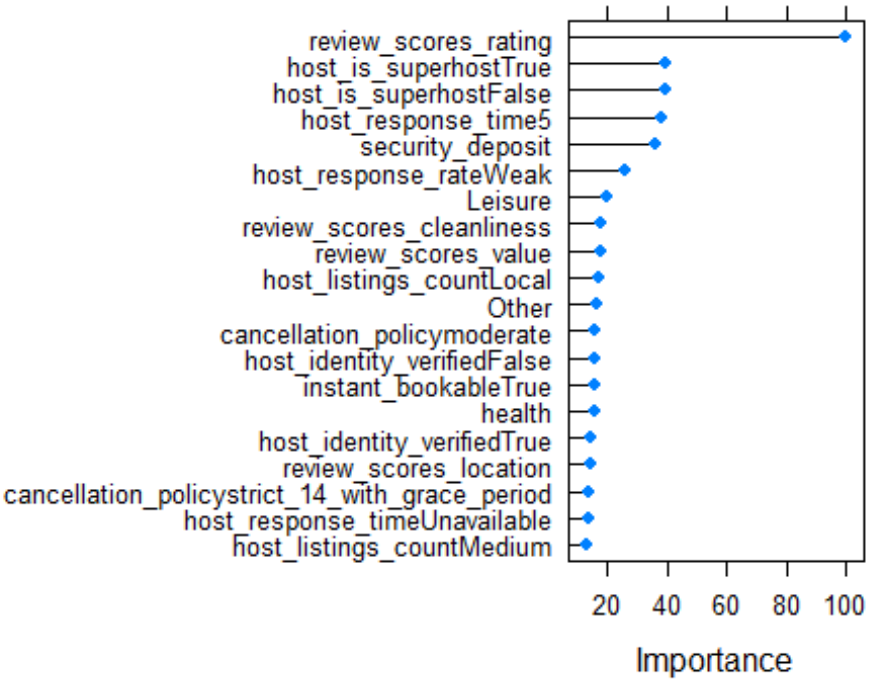



```
#Make predictions:
resultsBaggedTree2 <-
  fitBaggedTree2 %>%
  predict(austinTestSet, type='prob') %>%
  bind_cols(austinTestSet, predictedProb=.)


resultsBaggedTree2<- resultsBaggedTree2%>%
  mutate(predictedClass = as.factor(ifelse(resultsBaggedTree2$`1` > 0.35, 1,
0)))


resultsBaggedTree2 %>%
  xtabs(~predictedClass+ high_booking_rate, .) %>%
  confusionMatrix(positive = '1') #%>%

## Confusion Matrix and Statistics
##
##                high_booking_rate
## predictedClass    0    1
##              0 1230  130
##              1  310  350
##
##                Accuracy : 0.7822
##                  95% CI : (0.7635, 0.8)
##     No Information Rate : 0.7624
```

```
##     P-Value [Acc > NIR] : 0.01874
##
##                   Kappa : 0.4675
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.7292
##             Specificity : 0.7987
##          Pos Pred Value : 0.5303
##          Neg Pred Value : 0.9044
##              Prevalence : 0.2376
##          Detection Rate : 0.1733
##    Detection Prevalence : 0.3267
##       Balanced Accuracy : 0.7639
##
##        'Positive' Class : 1
##
```

Management Model:

```
#See the variables plotted by importance (according to the bagged tree):
plot(varImp(fitBaggedTree1), top=20)
```



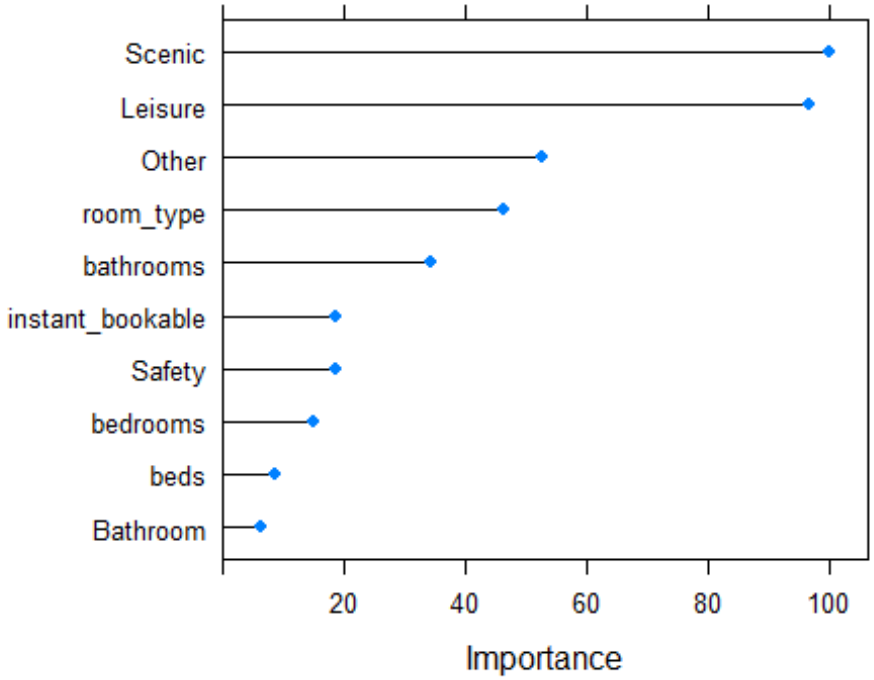

```
#Make predictions:
resultsBaggedTree1 <-
  fitBaggedTree1 %>%
  predict(austinTestSet, type='prob') %>%
  bind_cols(austinTestSet, predictedProb=.)
```

```
  resultsBaggedTree1 <- resultsBaggedTree1%>%
    mutate(predictedClass = as.factor(ifelse(resultsBaggedTree1$`1` > 0.35, 1
, 0)))

resultsBaggedTree1 %>%
  xtabs(~predictedClass+ high_booking_rate, .) %>%
  confusionMatrix(positive = '1') #%>%
```

```
## Confusion Matrix and Statistics
##
##                high_booking_rate
## predictedClass    0    1
##              0 1270  107
##              1  270  373
##
##                Accuracy : 0.8134
##                  95% CI : (0.7957, 0.8301)
##     No Information Rate : 0.7624
##     P-Value [Acc > NIR] : 1.89e-08
##
##                   Kappa : 0.5388
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7771
##             Specificity : 0.8247
##          Pos Pred Value : 0.5801
##          Neg Pred Value : 0.9223
##              Prevalence : 0.2376
##          Detection Rate : 0.1847
##    Detection Prevalence : 0.3183
##       Balanced Accuracy : 0.8009
##
##        'Positive' Class : 1
##
```

Upgrades and renovations:

```
plot(varImp(fitKNN), top=10)
```




```
resultsKNN <-
  fitKNN %>%
  predict(austinTestSet_f, type='prob') %>%
  bind_cols(austinTestSet_f, predictedProb=.)

resultsKNN<- resultsKNN%>%
  mutate(predictedClass = as.factor(ifelse(resultsKNN$`1` > 0.35, 1, 0)))

resultsKNN %>%
  xtabs(~predictedClass+high_booking_rate, .) %>%
  confusionMatrix(positive = '1')
```

```
## Confusion Matrix and Statistics
##
##                high_booking_rate
## predictedClass    0    1
##              0 1229  237
##              1  311  243
##
##                Accuracy : 0.7287
##                  95% CI : (0.7088, 0.748)
##     No Information Rate : 0.7624
##     P-Value [Acc > NIR] : 0.999793
##
##                   Kappa : 0.289
##
##  Mcnemar's Test P-Value : 0.001818
```

```
##
##             Sensitivity : 0.5062
##             Specificity : 0.7981
##          Pos Pred Value : 0.4386
##          Neg Pred Value : 0.8383
##              Prevalence : 0.2376
##          Detection Rate : 0.1203
##    Detection Prevalence : 0.2743
##       Balanced Accuracy : 0.6522
##
##        'Positive' Class : 1
##
```

Conclusion and Discussion

Austin is the fast-growing metropolitan region in the United States and is rated as one of the best regions in the country to invest and live. The pieces of evidence present in the case can prove this fact.

Business Case 1 - Initial acquisition of the houses

- It is best to consider investing in a property which is *available* to occupy all throughout the year and number of *bedrooms* and *bathrooms* are essential parameters in deciding on the property to choose to invest on. It is recommended to have 3 bed and 2 bath apartments. Another important aspect is the number of people that could be *accommodated* in the house. This business case successfully touches upon pre-investment features from a buyer's perspective to gain profits.

- The reason why we considered the availability as well in this case is because there are some properties that would not be available during some part of the year. Austin is visited more often for long weekends around major sport events and the music festivals. For the music festivals the bookings are made approximately a year ahead and sport events bookings are made 3 months ahead. So, we suggest the host buy a property which could be made available around these time spans.

Business Case 2

Business Case 2(1) - Pricing

While considering the prices for the properties, the factors that influenced the most were the *availability* of the property all throughout the year, the *cleaning fee* and the number of people the property *accommodates*. It is more surprising to see that these parameters had a higher influence than the number of *bedrooms* and *bathrooms*. Evidently, the people have higher importance for hygiene and cleanliness.

Business Case 2(2) - Management

- For this question, we did some research on how the houses are managed and we found out that there are companies out there who will manage your rental property for you. These are classified into professional management in our case. After the analysis we found out that professional management is not essential to look after the properties. What matters more is the *review scores rating*.

- Surprisingly it doesn't matter if a host is a *super host or not* while considering a property in Austin Market. However, *host response rate, host response time, host response rate* play a major role in deciding on the property. That means it would be great if the host would respond quickly

- The models we made supported our research questions as well as contradicted some of the expectations that we had. Some of the expectations were obvious such as you would expect a superhost to have a higher chance of getting a high booking rate as compared to normal hosts.

Business Case 3 - Upgrades and Renovations

- Our focus over here is to help the investor to further improve the bookings by making modifications to the property that they purchased.

- For that we made models based on the amenity categories that we made. We found out that the *scenic amenities*{Patio or Balcony, Outdoor Seating, Mountain View, etc} and *leisure* amenities{Private Pool, Outdoor BBQ, Netflix, Hammock, etc} were significantly important in our model result.

- Some of these amenities are really easy to upgrade to and have a good ROI. Our market research also corroborates these findings from the model. We know that Texan houses are large in size and these can be easily implemented.

Limitations :

- The data we used for our analysis did not contain a time series component(dates for customer booking). The findings and assumptions were made from online resources and locals at Austin about the high booking period of the year. Lack of time series component made us unable to endorse a model with evidence of high booking in the duration which was researched via external resources.

- Also, variables such as access, description, neighborhood overview contained different types of construction in each row. Generalizing categories for such varied fields is a big task. Therefore, we decided not to consider these variables in our model. Taking into consideration these variables may improve model performance. Furthermore, the variables having more than half missing values were unconsidered. Using some metrics to replace N/A values may have changed the complete distribution of variables. These were *monthly_price* and *interaction*. Again, having proper observations for these columns could have led to better results.

- Finally, using mode to replace missing values was our approach. As beginners we are unaware of the hidden intricacies of statistical metrics. The one which gave high performance was selected. This may have incorrectly replaced some of the missing values.

Future Scope :

- The further research can be improved in quality by overcoming the limitations faced. Comprehensive research on customer data with time series components can be merged with existing data models.

- Text mining, sentiment analysis on different columns having different text in each row can be performed to gain meaningful insights from unused columns. Example, access variables can detect if there is any correlation between ease of route to be taken to the property and high booked property. Lastly, in depth study of statistical metrics and its effect on models will better help us to replace missing values without focusing only on performance measures.

References

https://vacationidea.com/ideas/best-time-to-visit-austin-tx.html
https://patch.com/texas/eastaustin/austin-9th-best-nation-airbnb-operators-report
https://www.thebrokebackpacker.com/where-to-stay-in-austin-usa/#East%20Austin
https://www.thecrazytourist.com/25-best-things-austin-texas/
https://www.kaggle.com/clnguyen/austinairbnbs20191112
https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9 https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
https://machinelearningmastery.com/confusion-matrix-machine-learning/
https://en.wikipedia.org/wiki/Sensitivity_and_specificity