Data Mining and Predictive Analysis

Predicting Booking Rates of Airbnb Listings

Washington DC

"We, the undersigned, certify that the report submitted is our original work; all authors participated in the work in a substantial 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."

	Names of the signed team members
Contact Member	Dharmik Bhayani
Team Member 2	Seema Agarwal
Team Member 3	Jenil Kansara
Team Member 4	Khushboo Modi
Team Member 5	Jalvi Sheta

1. Executive Summary

The major part of our analysis focused on identifying the factors which can help investors achieve a high booking rate. Prediction was an integral part of our study. Accurate prediction would help an investor in focusing on properties that could give him a competitive edge. Our approach focused on looking at the data through a customer's eyes. We tried to include in our model all variables that a customer might deem important in choosing an Airbnb. We were able to decipher some interesting facts from our results.

Location does not play a significant role in getting a high booking rate. So for any existing property owner in DC, listing it as Airbnb would get more revenue, than renting the property to a tenant. And diversifying the investment is a good strategy to make sure your portfolio is profitable. Achieving Superhost Status will significantly increase your chances of high booking rate. While Airbnb has its own criteria for that status, things of utmost importance are catering and matching the user requirements, and maintaining a good image. Having a good response time, honouring all bookings made and great reviews will get you closer to superhost status and bring your listing to the top.

We have tried at every point to make sure that all our findings can be explained and it is not just a black box prediction methodology. This helped us in building a strong business case

with clearly explained variables and results. Still, the degree of reliance on the model would depend on what the business requirements are. Combining human judgement and the insights from the model for decision making would be an optimal solution.

2. Research Questions

1. What factors affected the booking rates of an Airbnb listing?

i) What kind of Airbnb - rooms, facilities, number of beds and bathrooms - have a higher booking rate?

ii) Do Airbnb hosts with multiple listings achieve better booking rates?

iii) What kind of services and facilities affect the booking rate of Airbnb?

2. What are the relevant statistics about the DC short term rental market?

i) Is there any clustering in type of airbnb in areas?

ii) Which areas have higher booking rates?

iii) What kind of people visit dc, and what is their purpose?

iv) What type of housing they might be looking for? In what price range?

v) Generally what is the trip duration/stay in dc?

We focussed mainly on the above two questions because our goal is to provide the investor with recommendations backed up with evidence. If we have knowledge about the factors which affect the booking rates of an Airbnb listing, then the investor would have a fair amount of idea about the amenities that hold most importance, type of properties to invest in, the areas which are more profitable than others. Secondly, understanding specifics about the Washington DC market will help in making right decisions while investing in DC. Some factors which are profitable in one market might not be of any significance in another market.

3. Methodology

For the first part of the project, which is for the whole US Airbnb market,the initial step was to prepare the data for analysis. The original dataset has more than 60 variables. We definitely needed to narrow it down. We selected variables first by asking ourselves what a customer would look for in an Airbnb. The list was long and not all data could be used directly. We needed to convert into a format that would make sense to us as well as the model. A lot of our time was taken up by data cleaning, transformations, null values handling and creating new variables . For data preprocessing, we imputed as well as transformed some of the variables:

1. Some entries for the number of bedrooms and bathrooms were missing in the dataset. The missing values accounted for approximately 0.1% of the total. So, the mean value of bedrooms and bathrooms was used to perform the imputation.

2. To consider Review Score into our model, it was important to categorize them. Therefore, we formed three different bins for review i.e. poor, okay and good. After looking at the distribution of review scores; review_scores_rating greater than 90 were categorized as good, less than 90 were categorized as okay, and where null values were present it was categorized as bad.

3. Response times within a day were categorized as good and others were categorized as bad.

4. When security fees and cleaning fees were Null in the dataset, it was imputed with zero. We assume that if the value is missing, it is zero otherwise it would have been mentioned.

5. Years Active was derived using the variable host_since. It was calculated by subtracting the current year from the year the host first listed a property on Airbnb.

6. The number of amenities is calculated from the amenities variable present in the dataset. The number of text entries separated by comma are counted and a new variable no_of_amenities is defined.

We used these variables to build predictive models. After trying logistic regression, random forest, and several other models, we found that xgboost gave the highest performance, with AUC of **0.93**.

```
train <- train %>% mutate(new = as.integer(`{randomControl}`/1000)) %>% filte
r(new == 107)

col_in_train <- c("id", "amenities","bathrooms", "bedrooms", "accommodates", "
cancellation_policy", "cleaning_fee", "extra_people", "host_is_superhost","ho
st_listings_count", "price", "review_scores_rating", "high_booking_rate", "ho
st_response_rate","host_response_time", "minimum_nights", "security_deposit",
"property_type", "latitude","longitude")

dc_data <- train[col_in_train]

dc_data <- dc_data %>% mutate(amenities = tolower(amenities))

must_have_amenities                          <-                          c("parking")
dc_data <- dc_data %>% bind_cols(as.data.frame(sapply(must_have_amenities, gr
epl, dc_data$amenities)))

recode_policy <- as.list(dc_data %>% group_by(cancellation_policy) %>% tally(
) %>% mutate(pct = n/sum(n)*100) %>% filter(pct < 10))[1]

apartment <- c('Apartment', 'Aparthotel', 'Boutique hotel', 'Condominium', 'H
otel',      'Loft',      'Serviced      apartment',      'Guest      suite')

house <- c('Bungalow', 'Villa', 'Castle', 'Cabin', 'Hut', 'Cottage', 'Barn', '
Tiny house', 'Bed and breakfast', 'Casa particular (Cuba)', 'Townhouse', 'Ear
th house', 'Chalet', 'Dome house', 'Guesthouse', 'House', 'In-law', 'Nature l
odge', 'Resort', 'Vacation home')

dc_data <- dc_data %>% mutate(high_booking_rate = as_factor(high_booking_rate
),
                                  no_of_amenities = sapply(strsplit(amenities
,                                ","),                                    length),
                                  cancellation_policy = if_else(cancellation_
```

```
policy        %in%        unlist(recode_policy),"Other",cancellation_policy),
                                  cancellation_policy = as_factor(cancellatio
n_policy),
                                  property_type = if_else(property_type %in% a
partment,"apartment",   if_else(property_type   %in%   house,"house","other")),
                                  property_type   =   as_factor(property_type),
                                  cleaning_fee = as.numeric(gsub('\\$|,', '',
cleaning_fee)),
                                  cleaning_fee = if_else(is.na(cleaning_fee),
0,cleaning_fee),
                                  extra_people = as.numeric(gsub('\\$|,', '',
extra_people)),
                                  price = as.numeric(gsub('\\$|,', '', price)
),
                                  derived_review = if_else(review_scores_rati
ng >= 90, "Good",if_else(review_scores_rating >0 & review_scores_rating < 90,
"Okay",                                                                  "Poor")),
                                  derived_review = if_else(is.na(derived_revi
ew),"Poor",derived_review),
                                  derived_review = as_factor(derived_review),
                                  host_response_rate = as.numeric(gsub('\\%|,
',                          '',                          host_response_rate)),
                                  host_response_rate = if_else(host_response_
rate                   >                   90,                   "Good","Bad"),
                                  host_response_rate = if_else(is.na(host_res
ponse_rate),"Bad",host_response_rate),
                                  host_response_rate = as_factor(host_respons
e_rate),
                                  host_response_time = if_else(host_response_
time %in% c("within an hour", "within a few hours","within a day"),"Good", "B
ad"),
                                  host_response_time = as_factor(host_respons
e_time),
                                  minimum_nights = if_else(minimum_nights <= 2
,                                "Good",                                  "Bad"),
                                  minimum_nights = as_factor(minimum_nights),
                                  security_deposit = as.numeric(gsub('\\$|,',
'',                                                  security_deposit)),
                                  security_deposit = if_else(is.na(security_d
eposit),0,security_deposit),
                                  host_is_superhost = if_else(is.na(host_is_s
uperhost),FALSE,host_is_superhost),
                                  bedrooms = if_else(is.na(bedrooms),mean(bed
rooms,                na.rm                =                TRUE),bedrooms),
                                  bathrooms = if_else(is.na(bathrooms),mean(b
athrooms,                na.rm                =                TRUE),bathrooms),
                                  host_listings_count = if_else(is.na(host_li
stings_count),mean(host_listings_count, na.rm = TRUE),host_listings_count))

## Warning: NAs introduced by coercion
```

```
remove_var                <-                c("amenities","review_scores_rating")
dc_data <- dc_data[!names(dc_data) %in% remove_var]
```

For the second part of our project we focus on the Washington DC market; the capital of the United States, a culturally rich, compact city with many monuments and museums. If we look at the data of visitors to DC, the majority come to the city with the purpose of Vacation. And with DC being the 10th biggest visitor market in the US, it is worthwhile to understand the market better. With the knowledge we gained from the Kaggle competition and the market research, we moved on to explore the data for DC:

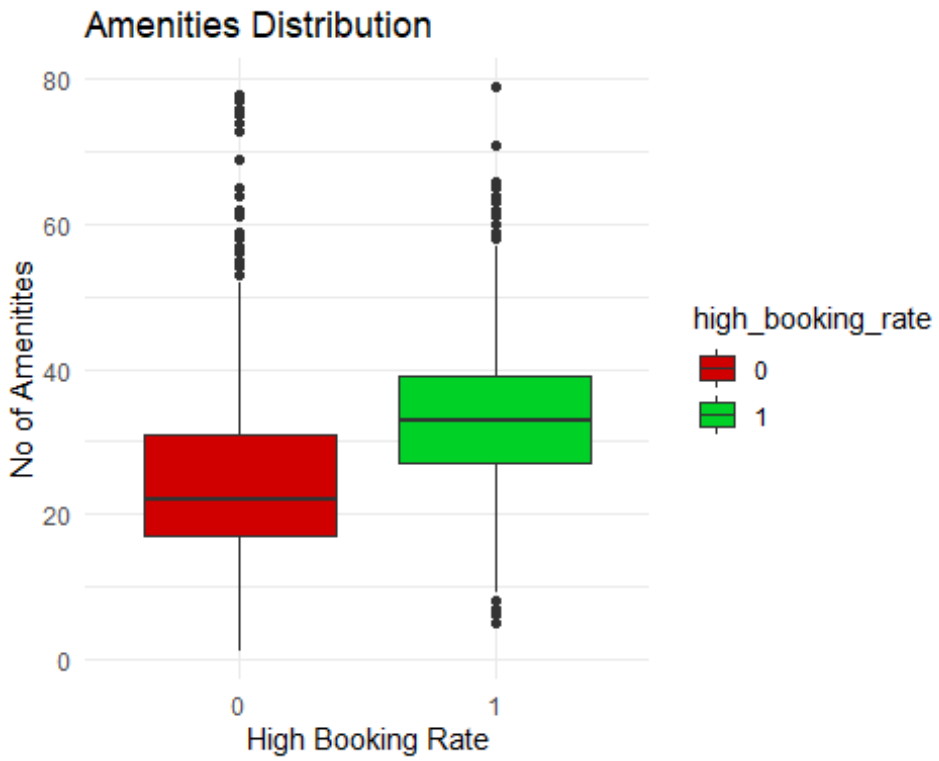
1. Price is of course a necessary variable to explore. There is a large section of people that chooses Airbnbs for their low rates. Do high priced Airbnbs do as well as low priced Airbnbs? We see from the boxplots that Airbnbs with unreasonably high prices will not achieve a high booking rate. An interesting thing to note is that the one priced at 10000 is a typing error, but it is the listing of Veep Suite at Hamilton Hotel Washington D.C, a replica of the set of the Emmy winning HBO series VEEP.

```
dc_data %>% filter(price <= 1000) %>% ggplot(aes(y = price, x = high_booking_
rate, fill = high_booking_rate)) + geom_boxplot() + xlab("High Booking Rate")
+ ylab("Rental Price") + ggtitle("Rental Price Distribution")+scale_fill_manu
al(values=c( "#d10000","#00d126")) + theme_minimal()
```

2. When there are a lot of homes to choose from, would the number of amenities influence the customer? This question motivated us to plot this graph. The graph shows the distribution of the number of amenities divided by high booking rate. From this stacked

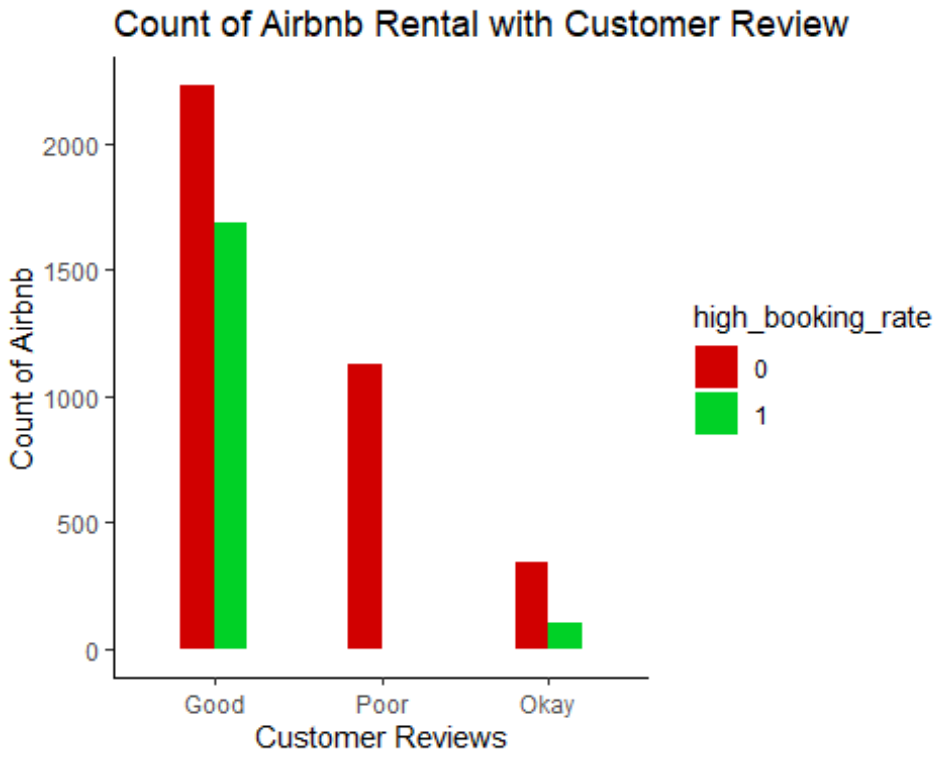
histogram, we can see having high amenities influences the listings to have a high booking rate.

```
dc_data %>% ggplot(aes(y = no_of_amenities, x = high_booking_rate, fill = hig
h_booking_rate)) + geom_boxplot() + xlab("High Booking Rate") + ylab("No of A
menitites") +theme_minimal() + scale_fill_manual(values=c( "#d10000","#00d126
")) + ggtitle("Amenities Distribution")
```



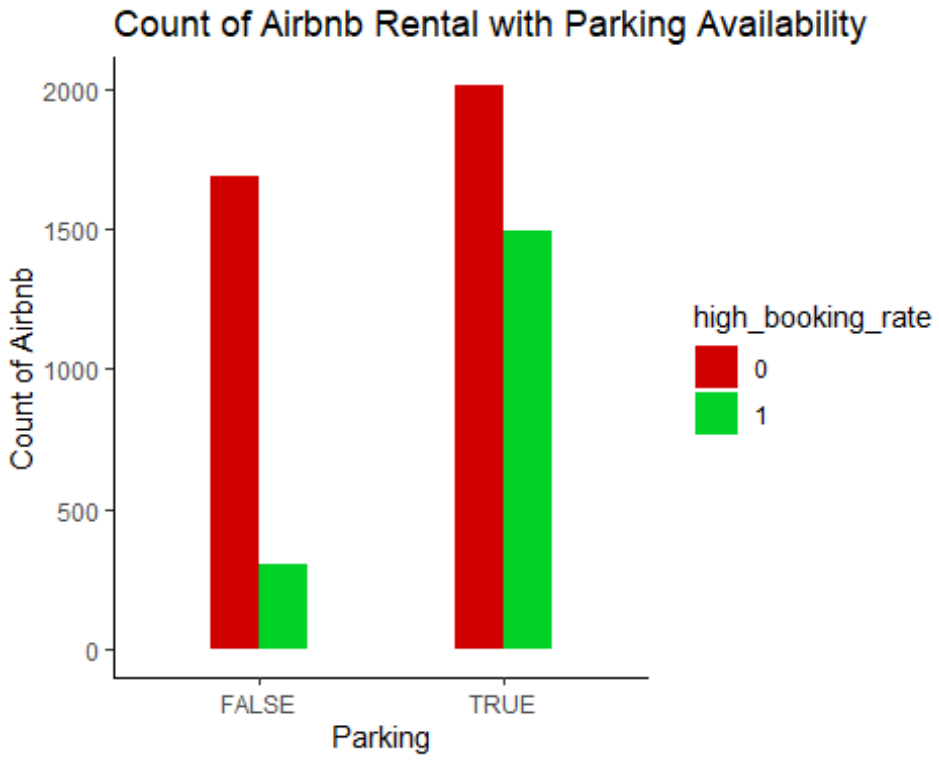
3. Having good reviews is always good for business. And we reach the same conclusion that we had for parking. Even if good reviews won't guarantee a greater number of bookings, having poor reviews makes you susceptible to low booking rates.

```
ggplot(data.frame(dc_data %>% xtabs(~high_booking_rate+derived_review, .)), a
es(fill = high_booking_rate, y = Freq, x = derived_review)) + geom_bar(positi
on = "dodge", stat = "identity", width = 0.4) + xlab("Customer Reviews") + yl
ab("Count of Airbnb") + ggtitle("Count of Airbnb Rental with Customer Review"
) + theme_classic() + scale_fill_manual(values=c( "#d10000","#00d126"))
```

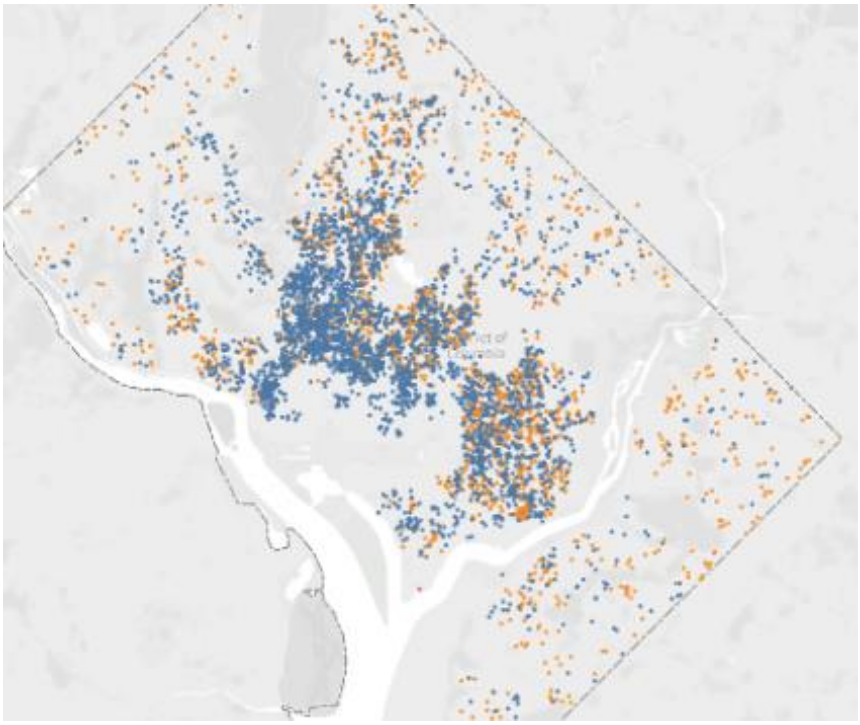


4. DC has 23 million annual visitors and 20 million of them are domestic travellers. So we thought parking could be a good variable to explore. We found that having parking space won't guarantee you a high booking rate but not having them makes you more likely to have low booking rates. So this is an approach of avoiding certain situations that will make the listing have a low booking rate.

```
ggplot(data.frame(dc_data %>% xtabs(~high_booking_rate+parking, .)), aes(fill
= high_booking_rate, y = Freq, x = parking)) + geom_bar(position = "dodge", s
tat = "identity", width = 0.4) + xlab("Parking") + ylab("Count of Airbnb") + g
gtitle("Count of Airbnb Rental with Parking Availability") + theme_classic() +
scale_fill_manual(values=c( "#d10000","#00d126"))
```

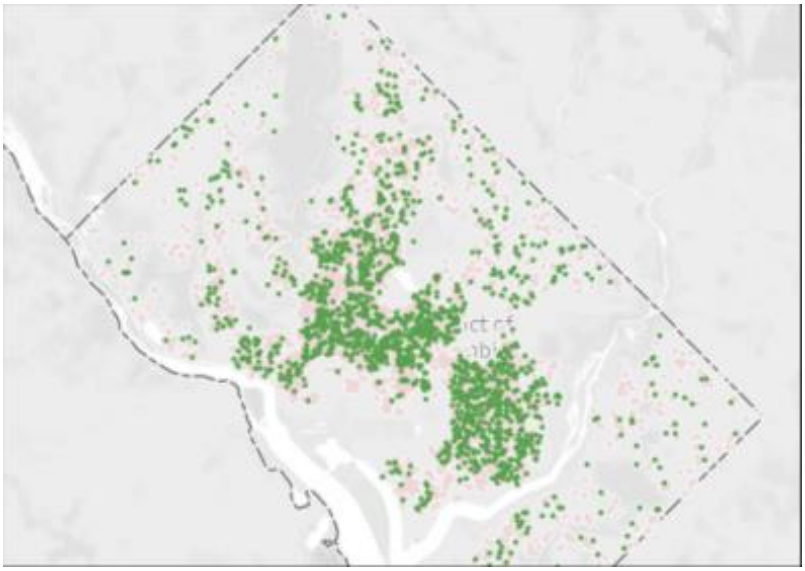


5. It is important for an investor to know what kind of property he should buy and where. So through this visualization for the Washington DC market, we found out that apartments are concentrated in the central part of the market whereas houses and other property types are mostly found in the suburbs of the city.

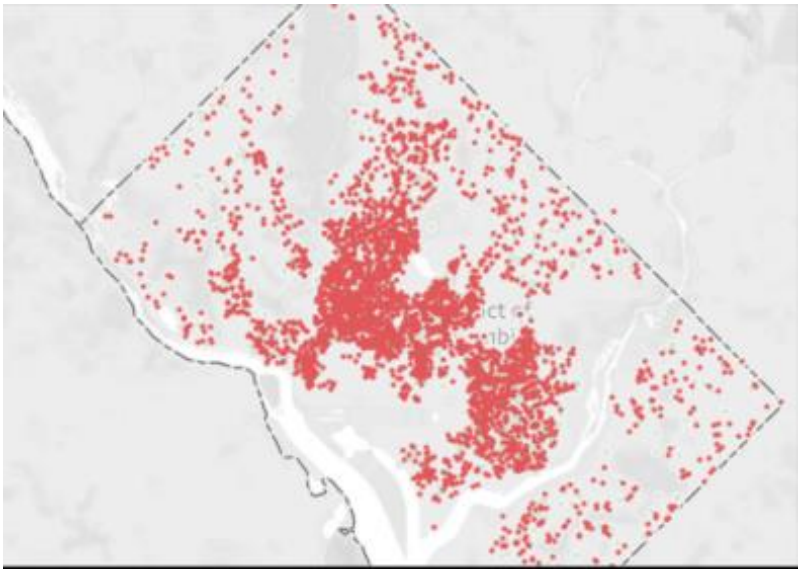


Distribution of House Types

6. We had an initial hypothesis that location would affect booking rates. But here we can see that there is no cluster of Airbnbs with a high or low booking rate in the market. There is no particular neighborhood which can be thought of as an area of high booking rate Airbnbs.



High Booking Rate



Low Booking Rate

7. In following types of combinations, atleast 30% of properties achieve high booking rate For apartments: Studio, 1 Bed and 1 Bath, 2 Bed and 1/2 Bath, 3 Bed and 2.5 Bath For houses: 1 Bed and 1/1.5 Bath, 2 Bed and 1/2 Bath, 3 Bed and 2/2.5 Bath

```
dc_data %>% select(property_type, bedrooms,bathrooms, high_booking_rate) %>% g
roup_by(property_type, bedrooms, bathrooms, high_booking_rate) %>% tally() %>
% spread(high_booking_rate,n) %>% rename("Low_Booking_Rate" = "0", "High_Book
ing_Rate" = "1") %>% mutate(Percentage = High_Booking_Rate/(Low_Booking_Rate+
High_Booking_Rate)*100) %>% arrange(desc(property_type,Percentage)) %>% filte
r(Low_Booking_Rate+High_Booking_Rate >= 100)
```

```
##           #           A           tibble:           8           x           6
##    #    Groups:             property_type,    bedrooms,    bathrooms    [9]
##   property_type bedrooms bathrooms Low_Booking_Rate High_Booking_Rate Perc
entage
##   <fct>            <dbl>     <dbl>            <int>              <int>      <
dbl>
## 1 house                1       1                482                302       3
8.5
## 2 house                1       1.5              121                 68       3
6.0
## 3 house                2       1                 81                 48       3
7.2
## 4 house                3       2.5               96                 42       3
0.4
## 5 apartment            0       1                323                147       3
1.3
## 6 apartment            1       1               1369                690       3
3.5
## 7 apartment            2       1                216                124       3
6.5
## 8 apartment            2       2                202                 67       2
4.9
```

---

4. Modeling

Modeling for the DC market, we developed a Logistic Regression model for Exploratory purposes as the importance of the variables can be explained and XGBoost for prediction because even though it does not explain variables, it gives more accurate predictions.

Specificity (True Negative Rate) is more important as investing in a unit that is not highly bookable would be costly. Here, we try to decrease the percentage of false positives.

There are multiple ways of selecting a cut-off.

Cut off <= 0.4

There will be more false positive cases here, which means that the investor will invest in more Airbnbs which will have low booking rates, which would be costly.

Cut off >= 0.6

If the investor is risk averse, then it would be an ideal case to pursue. Here, there would be more missed opportunities, but the amount of capital lost in risky properties investment will be lower. This is a risk averse situation.

Cut off = 0.5

This is a risk neutral scenario, which is a balance between the scenario one and two.

Hence the cutoff will be modified depending on how much risk an investor is willing to take.

```
set.seed(333)
dc_train                        <-                        sample_frac(dc_data,0.8)
dc_test <- dplyr::setdiff(dc_data, dc_train)

dc_logistic <- glm(high_booking_rate ~ . -id -accommodates -latitude - longit
ude, family = binomial(), data = dc_train)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(dc_logistic)

##
##                                                                        Call:
##  glm(formula = high_booking_rate ~ . - id - accommodates - latitude -
##                longitude,   family   =   binomial(),   data   =   dc_train)
##
##                              Deviance                                 Residuals:
##           Min                1Q       Median               3Q              Max
##    -2.17964       -0.69051       -0.00016          0.73548           2.71242
##
##                                                                 Coefficients:
##                                                Estimate Std. Error z valu
e
## (Intercept)                                   -2.581e+00  1.924e-01 -13.4
13
## bathrooms                                     -4.240e-01  9.785e-02  -4.3
33
## bedrooms                                       1.419e-01  6.788e-02   2.0
91
## cancellation_policymoderate                    6.606e-01  1.088e-01   6.0
70
## cancellation_policystrict_14_with_grace_period  6.005e-01  1.152e-01   5.2
13
## cancellation_policyOther                       8.683e-01  8.262e-01   1.0
51
## cleaning_fee                                  -2.632e-03  1.127e-03  -2.3
35
## extra_people                                   1.997e-03  1.681e-03   1.1
88
## host_is_superhostTRUE                          1.042e+00  8.916e-02  11.6
87
## host_listings_count                           -4.551e-03  2.116e-03  -2.1
51
## price                                         -8.925e-04  4.236e-04  -2.1
07
## host_response_rateBad                          1.131e-01  1.619e-01   0.6
99
## host_response_timeBad                         -1.121e+00  1.864e-01  -6.0
12
```

```
## minimum_nightsGood                                  6.319e-01  9.448e-02   6.6
88
## security_deposit                                   -4.131e-04  1.570e-04  -2.6
31
## property_typehouse                                  5.887e-02  9.181e-02   0.6
41
## property_typeother                                 -7.378e-01  8.243e-01  -0.8
95
## parkingTRUE                                         2.974e-01  1.050e-01   2.8
33
## no_of_amenities                                     5.385e-02  5.003e-03  10.7
64
## derived_reviewPoor                                 -1.657e+01  1.942e+02  -0.0
85
## derived_reviewOkay                                 -3.708e-01  1.458e-01  -2.5
44
##                                                                          Pr(>|z|)
##  (Intercept)                                                           < 2e-16 ***
##  bathrooms                                                            1.47e-05 ***
##  bedrooms                                                               0.03653 *
##  cancellation_policymoderate                                          1.28e-09 ***
##      cancellation_policystrict_14_with_grace_period                1.85e-07      ***
##  cancellation_policyOther                                                0.29328
##  cleaning_fee                                                          0.01956 *
##  extra_people                                                            0.23475
##  host_is_superhostTRUE                                                < 2e-16 ***
##  host_listings_count                                                    0.03148 *
##  price                                                                  0.03513 *
##  host_response_rateBad                                                   0.48479
##  host_response_timeBad                                                1.83e-09 ***
##  minimum_nightsGood                                                   2.26e-11 ***
##  security_deposit                                                      0.00852 **
##  property_typehouse                                                      0.52140
##  property_typeother                                                      0.37079
##  parkingTRUE                                                           0.00461 **
##  no_of_amenities                                                      < 2e-16 ***
##  derived_reviewPoor                                                      0.93201
##  derived_reviewOkay                                                    0.01096 *
##                                                                                ---
##  Signif.  codes:    0  '***'  0.001  '**'  0.01  '*'  0.05  '.'  0.1  ' '  1
##
##   (Dispersion   parameter   for   binomial   family   taken   to   be   1)
##
##            Null   deviance:   5573.1     on   4393     degrees   of   freedom
##   Residual   deviance:   3639.2      on   4373      degrees   of   freedom
##                                  AIC:                                    3681.2
##
## Number of Fisher Scoring iterations: 17
```

```
predict_dc_logistic <- predict(dc_logistic, dc_test, type = "response") %>%
  bind_cols(dc_test,          predictionProb          =          .)          %>%
  mutate(predictedClass = if_else(predictionProb >= 0.5,1,0))

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

##            Confusion            Matrix            and            Statistics
##
##                                                          high_booking_rate
##         predictedClass                          0                          1
##                                                    0    644    124
##                                                    1    112    218
##
##                                                 Accuracy    :    0.7851
##                                             95%   CI    :   (0.7596,   0.809)
##                            No      Information      Rate      :      0.6885
##                       P-Value     [Acc     >     NIR]     :     5.834e-13
##
##                                                 Kappa    :    0.494
##
##                 Mcnemar's        Test        P-Value        :        0.474
##
##                                            Sensitivity    :    0.6374
##                                            Specificity    :    0.8519
##                                       Pos    Pred    Value    :    0.6606
##                                       Neg    Pred    Value    :    0.8385
##                                                Prevalence    :    0.3115
##                                          Detection    Rate    :    0.1985
##                            Detection       Prevalence       :       0.3005
##                                    Balanced     Accuracy     :     0.7446
##
##                                       'Positive'     Class     :     1
##

set.seed(2020)

dc_xgboost <- train(high_booking_rate ~ . -id -bedrooms -latitude - longitude
, data=dc_train, method='xgbTree', trControl=trainControl(method='cv', number
=10))

predict_dc_xgboost                                                    <-
  dc_xgboost                                                          %>%
  predict(dc_test,                      type='prob')                  %>%
  bind_cols(dc_test,                  predictedProb=.$"1")            %>%
  mutate(predictedClass = if_else(predictedProb >= 0.5,1,0))

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

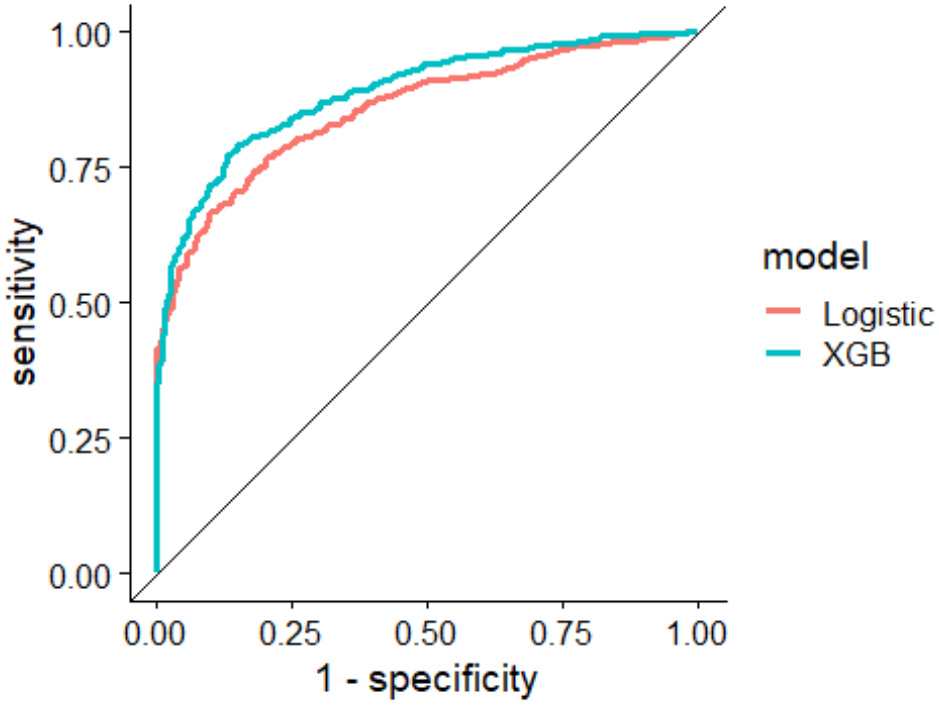
```
##            Confusion            Matrix            and            Statistics
##
##                                                          high_booking_rate
##        predictedClass                          0                         1
##                                                    0    657    104
##                                                  1        99    238
##
##                                                 Accuracy    :    0.8151
##                                          95%   CI   :   (0.7909,   0.8377)
##                            No      Information      Rate      :      0.6885
##                          P-Value      [Acc      >      NIR]      :      <2e-16
##
##                                                 Kappa    :    0.5672
##
##                 Mcnemar's        Test        P-Value        :        0.7789
##
##                                             Sensitivity    :    0.6959
##                                             Specificity    :    0.8690
##                                       Pos    Pred    Value    :    0.7062
##                                       Neg    Pred    Value    :    0.8633
##                                                Prevalence    :    0.3115
##                                          Detection    Rate    :    0.2168
##                            Detection       Prevalence       :       0.3069
##                                    Balanced     Accuracy     :      0.7825
##
##                                       'Positive'     Class     :      1
##
```

```
logistic     <-     predict(dc_logistic,dc_test,     type='response')     %>%
  bind_cols(dc_test,PredictedProb=                    .)                   %>%
  mutate(model                          =                          "Logistic")

xgb        <-        predict(dc_xgboost,dc_test,        type='prob')        %>%
  bind_cols(dc_test,PredictedProb=                  .$"1")                  %>%
  mutate(model                            =                               "XGB")

modelAll               <-               bind_rows(logistic,               xgb)

modelAll                                                                     %>%
  group_by(model)                                                            %>%
  roc_curve(truth      =      high_booking_rate,      PredictedProb)         %>%
  ggplot(aes(x  =  1  -  specificity,  y  =  sensitivity,  color  =  model))  +
  geom_line(size                    =                    1.1)                  +
  geom_abline(slope    =    1,    intercept    =    0,    size    =    0.4)    +
  coord_fixed()                                                              +
  theme_cowplot()
```




```
modelAll                                                                %>%
  group_by(model)                                                       %>%
  roc_auc(truth       =       high_booking_rate,       PredictedProb)       %>%
  arrange(desc(.estimate))
```

```
##           #           A           tibble:           2           x           4
##               model                       .metric     .estimator     .estimate
##         <chr>            <chr>         <chr>                                <dbl>
##   1   XGB                   roc_auc   binary                               0.891
## 2 Logistic roc_auc binary         0.859
```

5. Results and Findings

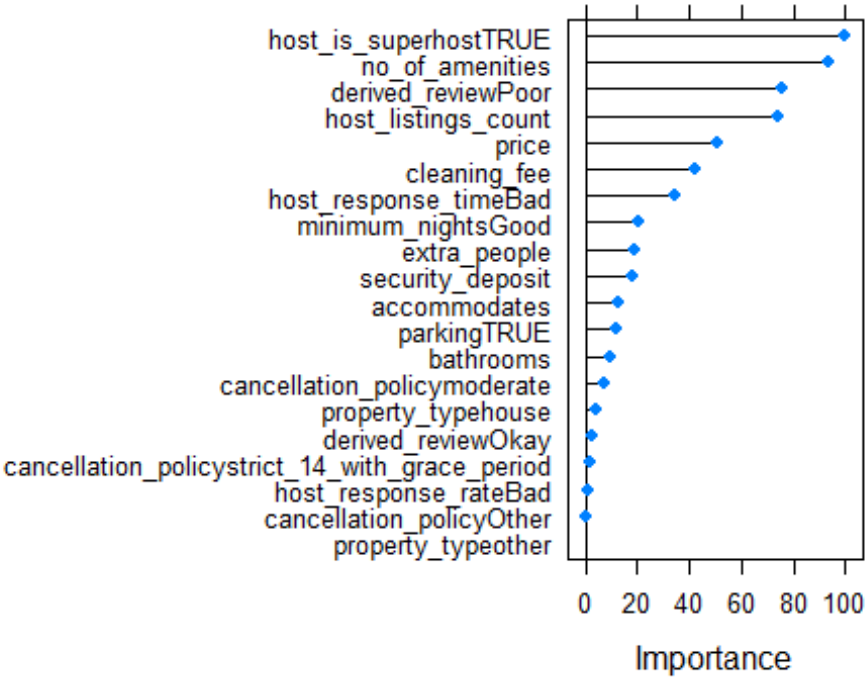
We were able to answer almost all of the questions we had set out to answer. We find that while buying the property, the combinations given below have around 30% chance of achieving high booking rate, all others have less than 20% (And as previously seen location has no effect on booking rate); Property Type: Central DC: Apartments with parking Suburbs: Houses with parking Bedrooms and Bathroom: For apartments: Studio, 1 Bed and 1 Bath, 2 Bed and 1/2 Bath, 3 Bed and 2.5 Bath For houses: 1 Bed and 1/1.5 Bath, 2 Bed and 2/2.5 Bath, 3 Bed and 2/2.5 Bath This makes sense as most people visit DC for vacation purposes, stay for an average of 3 nights and consist of small families. Apartments and houses of these sizes, which do not charge exorbitant prices, suffice their purpose. It would also be good to buy multiple properties and turn them into Airbnbs as we find that host with multiple listings have a higher chance of getting bookings. It is always good to diversify as it decreases the chance of loss.

To support our recommendation, below mentioned are the key variables that we found, converted to make the business decision making process a bit easier.

- If the price is increased by $20, then the odds of getting a high booking rate is decreased by 18%, keeping everything else as constant.
- For the response time, the odds of a host who takes more than a day to reply having a high booking rate are 70% lower than the odds of an Airbnb who takes less than a day to reply, keeping everything else constant.
- The odds of a superhost having a high booking rate are 80% higher than the odds of a non-superhost, keeping everything else as constant.
- Increasing the number of people staying in the rental by 2 is associated with an increase in the odds of a high booking rate by 28%, keeping everything else as constant.
- The odds of an Airbnb with parking having a high booking rate are 30% higher than the odds of an Airbnb with no parking facility, keeping everything else constant.
- The odds of an Airbnb property classified as a house having a high booking rate are 11% higher than the odds of other types of Airbnb properties, keeping everything else constant.

We now know which variables are important but how much importance should be given to each of them? And what should we keep in mind about them? Why are they important? The graph below shows the variable importance given by our XGBoost prediction model. While being a superhost has maximum weightage, that is not in our hands. Instead, an Airbnb host should focus on aggregating good reviews by providing excellent customer service, keeping the price competitive with the other listings in the area, responding to queries within a day and charging a minimum cleaning and security fee. The reason these variables have high importance is because following these will lead you one step closer to being a superhost.

```
plot(varImp(dc_xgboost), top=20)
```



Conclusion and Limitations

Through this project, we were able to get several interesting insights about the Airbnb market in DC. Below we will summarize the answers to the questions that we wished to answer at the beginning of the project: Maximum people visit DC for Vacations and Business purposes with an average trip duration of 3 nights and prefer an apartment or a house for staying. The properties having more number of amenities along with 1 to 3 beds and 1 to 2.5 baths have better booking rate. Moreover, the hosts who are superhost, have good reviews, keep competitive prices, quick response rate and multiple rentals have more chances of getting a high booking rate on their properties.

However in this project, we faced a limitation of not having a time series data for the property market of DC. If we have data which explains the growth of properties in particular areas over time then it could help in improving the predictions significantly. A future research could be made to determine what feature is the correct choice of measure for investment. As one of our observations suggests the Veep Suite at Hamilton Hotel Washington D.C which is priced at $10,000 per night has a low booking rate, but it could lead to high profits even with few bookings around the year. Hence, to know the previous statistics about revenue generated from a property would help to predict if investment will lead to desired return of investment.

References

1. Washington, DC Visitor Research. (2019, October 31). Retrieved May 7, 2020, from https://washington.org/press/dc-information/washington-dc-visitor-research

2. Strupp, J. (2019, April 4). Single-family homes take up a lot of space in the District. Retrieved May 7, 2020, from https://ggwash.org/view/71576/heres-how-much-of-dcs-housing-consists-of-single-family-homes

3. Washington, DC Facts. (2020, March 31). Retrieved May 7, 2020, from https://washington.org/dc-information/washington-dc-facts

4. DC Short Term Rental Laws: What's Changing In October 2019 | Nomadic Real Estate | Markets Insider. (n.d.). Retrieved May 7, 2020, from https://markets.businessinsider.com/news/stocks/dc-short-term-rental-laws-what-s-changing-in-october-2019-nomadic-real-estate-1028578319