- PREDICTIVE MODEL REPORT-

Index

1-DATA ANALYSIS PROCESS

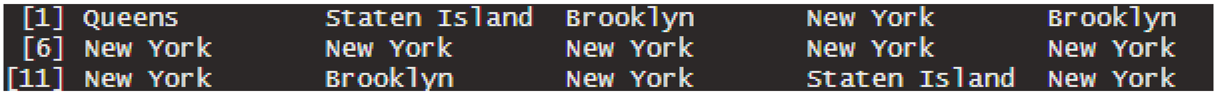
Cleaning the data

Eliminating unnecessary variables beforehand

Since the data is huge and extremely complex, I wanted to decrease this complexity through eliminating some variables apparently not related to price prediction such as country, country_code, jurisdiction_names, host_about etc.

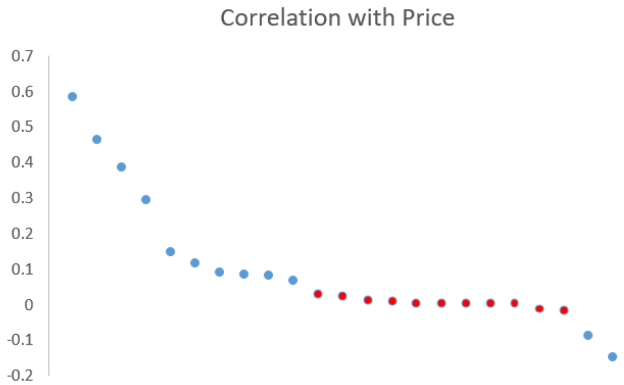
Location related redundancies

Airbnb data is comprised of dozens of variable as to 'location'. And to get a more concise data to train, I explored each location variable and held only ones that I think I can utilize. Accordingly, I examined their content based on head() and str() functions. Subsequently I eliminated location variables with high NA's and factor levels. Furthermore, columns with irrelevant information such as 'city' were also eliminated. To illustrate, here is the first 15 rows of 'city' variable:



```
 [1] Queens          Staten Island   Brooklyn        New York        Brooklyn
 [6] New York        New York        New York        New York        New York
[11] New York        Brooklyn        New York        Staten Island   New York
```

Elimanated variables due to low correlation

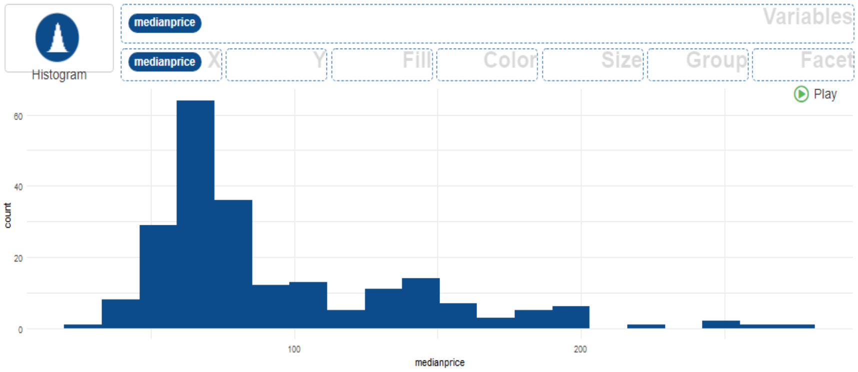


Finally, I go through the correlation between price and the remaining numeric variables in the data. Then, I spot the variables between 1% and -1% and consider them irrelevant. Thanks to these 3 phases, I shrink the main Airbnb file from 91 columns to 33 columns which means an approximate 2/3 reduction in size.

Fine-tuning NA's

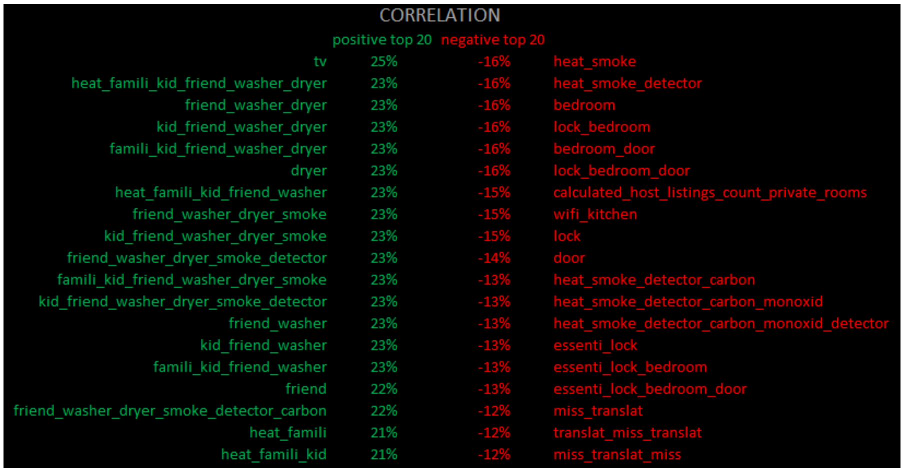
The grand NA issue in the data is related to 'Cleaning Fee' and 'Security Deposit' variables. Instead of imputing NA's with caret or mice packages, I decided to convert both NA's to '0'. Since these two variables represent a price, I translated NA directly meaning to '0'. Besides, some factor variables containing True, False booleans have also some NA's and I converted them into 1-0 format through using ifelse() and removed NA's.

Binning



Some significantly important variables such as 'neighbourhood_cleansed has 219 levels and I know that it prevents many predictive models from making calculations properly. Besides, thanks to binning, I have a chance to increase variables' performances through redesigning them based on median or mean price levels. Consequently, I handled these factor variables and redesigned them based on binning. I utilized esquisse package's esquiesser function to build distribution charts. After determining necessary binning points, I created new factor variables by using ifelse() function.

Text Mining (Amenity Variable)



CORRELATION

	positive top 20	negative top 20	
tv	25%	-16%	heat_smoke
heat_famili_kid_friend_washer_dryer	23%	-16%	heat_smoke_detector
friend_washer_dryer	23%	-16%	bedroom
kid_friend_washer_dryer	23%	-16%	lock_bedroom
famili_kid_friend_washer_dryer	23%	-16%	bedroom_door
dryer	23%	-16%	lock_bedroom_door
heat_famili_kid_friend_washer	23%	-15%	calculated_host_listings_count_private_rooms
friend_washer_dryer_smoke	23%	-15%	wifi_kitchen
kid_friend_washer_dryer_smoke	23%	-15%	lock
friend_washer_dryer_smoke_detector	23%	-14%	door
famili_kid_friend_washer_dryer_smoke	23%	-13%	heat_smoke_detector_carbon
kid_friend_washer_dryer_smoke_detector	23%	-13%	heat_smoke_detector_carbon_monoxid
friend_washer	23%	-13%	heat_smoke_detector_carbon_monoxid_detector
kid_friend_washer	23%	-13%	essenti_lock
famili_kid_friend_washer	23%	-13%	essenti_lock_bedroom
friend	22%	-13%	essenti_lock_bedroom_door
friend_washer_dryer_smoke_detector_carbon	22%	-12%	miss_translat
heat_famili	21%	-12%	translat_miss_translat
heat_famili_kid	21%	-12%	miss_translat_miss

As I explored amenities, there were loads of feature information regarding each rental so I decided to extract these information by employing text mining packages rweka, quanteda and reshape2. Consequently, I obtained a massive data with 352 variables. Afterward, I measured their correlation with price and filtered most important variables. Further, I also created a new variable by aggregating highly positive and highly negative correlated variables. Finally, I consolidated the text mined amenity data with the main data.

Creating Derivative Variables

On top of existent variables, I also created new variables by multiplying certain variables that has significant relationship with price. So I created 3 different derivative variables by multiplying different combinations of variables such as accommodation, bathroom, bedroom, cleaning fee, neighborhood cleansed, bed type and room type. I transformed these factor variables into bins with median price so that they can be directly multiplied.

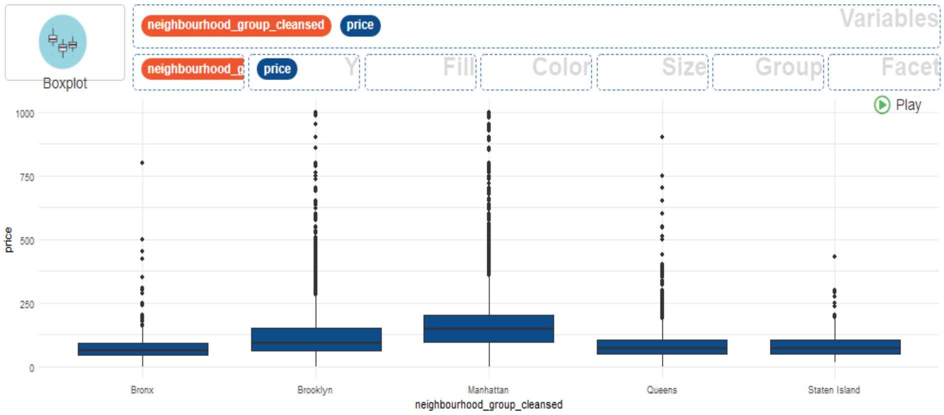
Prediction

For my final prediction I utilized random forest. Since it takes noticeable time to perform, I measured my rmse progress with basic decision tree model. Besides, I challenged and benchmarked my random forest model with gradient boosting.

2-What went wrong?

Outlier Removal

At the beginning I thought it would be logical to remove some outliers from data so I created box plots showing data with regards to neighborhood and price. Accordingly I eliminated some outliers such as prices above $500 in Queens etc. However, this process ended up with a gross variance between my train-test rmse and submission rmse.



Weekly Price (filling NA's with Decision Tree)

Weekly price seems the most valuable data in terms of its proximity to real price variable, yet only approximately 10% of it was available. I built a separate internal decision tree model to predict the remaining 90% instead of automatically imputing with caret or mice packages. Although I thought it went well, once I submit the prediction for real submission data, it again gave a higher rmse compared to that of my train/test data.

Error - Factor has new levels

At first, I decided to proceed with Zipcode variable as being my major location predictor. However, there were some other Zipcode information on the submission data so that my model did not work. Consequently I switched Zipcode variable with neighborhood_cleansed.

3- KEY TAKEAWAYS FROM THE KAGGLE EXPERIENCE

With this competition, I found ample opportunities to practice the fundamental teachings of Framework class. I solidified my skills in cleaning, organizing data and predictive modeling. Moreover, this challenged urged me to do my own research about text mining and binning in hopes of increasing the accuracy of my model. On the flip side, I learnt to cope with several format of errors due to particular reasons. Besides, I clearly realized that starting into an analysis without a proper plan or design, is extremely time-consuming and exhausting.

4- DETAILED CODE BREAKDOWN (56.9 RMSE)

Index

UPLOADING THE DATA

```
library(dplyr); library(readr);library(caret);library(stringr);
library(ggplot2);library(qdap);library(dplyr);library(tm);library(wordcloud);library(plotrix)
library(dendextend);library(ggplot2);library(ggthemes);library(RWeka)
library(reshape2);library(quanteda)

setwd('C:/Users/USER01/Desktop/R/Kaggle')
data_main = read.csv('analysisData.csv')
dataq=data_main
```

CLEANING THE DATA

# First glance removals using head()

```
dataq$zipcode = NULL;dataq$smart_location = NULL;dataq$transit = NULL
dataq$city = NULL;dataq$street = NULL;dataq$host_neighbourhood = NULL
dataq$host_location = NULL;dataq$neighbourhood = NULL;dataq$name = NULL
dataq$space = NULL;dataq$notes = NULL;dataq$summary = NULL;dataq$interaction = NULL
dataq$house_rules = NULL;dataq$host_since = NULL;dataq$neighborhood_overview = NULL
```

#Quantifying Factor Variables while converting NA's to 0

```
dataq$host_has_profile_pic = as.character(dataq$host_has_profile_pic)
dataq$host_has_profile_pic = ifelse(dataq$host_has_profile_pic == 't','1','0')
dataq$host_has_profile_pic = as.factor(dataq$host_has_profile_pic)
dataq$host_has_profile_pic = as.factor(dataq$host_has_profile_pic)

dataq$host_is_superhost = as.character(dataq$host_is_superhost)
dataq$host_is_superhost = ifelse(dataq$host_is_superhost == 't','1','0')
dataq$host_is_superhost = as.factor(dataq$host_is_superhost)
dataq$host_is_superhost = as.numeric(dataq$host_is_superhost)

dataq$instant_bookable = as.character(dataq$instant_bookable)
dataq$instant_bookable = ifelse(dataq$instant_bookable == 't','1','0')
dataq$instant_bookable = as.factor(dataq$instant_bookable)
dataq$instant_bookable = as.factor(dataq$instant_bookable)

dataq$require_guest_profile_picture = as.character(dataq$require_guest_profile_picture)
dataq$require_guest_profile_picture = ifelse(dataq$require_guest_profile_picture == 't','1','0')
dataq$require_guest_profile_picture = as.factor(dataq$require_guest_profile_picture)
dataq$require_guest_profile_picture = as.factor(dataq$require_guest_profile_picture)

dataq$require_guest_phone_verification = as.character(dataq$require_guest_phone_verification)
dataq$require_guest_phone_verification = ifelse(dataq$require_guest_phone_verification == 't','1','0')
```

```
dataq$require_guest_phone_verification = as.factor(dataq$require_guest_phone_verification)
dataq$require_guest_phone_verification = as.factor(dataq$require_guest_phone_verification)

dataq$is_location_exact = as.character(dataq$is_location_exact)
dataq$is_location_exact = ifelse(dataq$is_location_exact == 't','1','0')
dataq$is_location_exact = as.factor(dataq$is_location_exact)
dataq$is_location_exact = as.factor(dataq$is_location_exact)

dataq$host_identity_verified = as.character(dataq$host_identity_verified)
dataq$host_identity_verified = ifelse(dataq$host_identity_verified == 't','1','0')
dataq$host_identity_verified = as.factor(dataq$host_identity_verified)
dataq$host_identity_verified = as.factor(dataq$host_identity_verified)

dataq$host_response_rate = as.character(dataq$host_response_rate)
dataq$host_response_rate = str_replace_all(dataq$host_response_rate,'N/A','0%')
dataq$host_response_rate = str_replace_all(dataq$host_response_rate,'%','')
dataq$host_response_rate = as.integer(dataq$host_response_rate)

dataq$cancellation_policy = as.character(dataq$cancellation_policy)
dataq$cancellation_policy = ifelse(dataq$cancellation_policy ==
'flexible','1',ifelse(dataq$cancellation_policy == 'moderate','2','3'))
dataq$cancellation_policy = as.factor(dataq$cancellation_policy)
```

# Low correlated variables are removed

```
# correlation matrix for numerics
#correlation_matrix = data.frame(cor(dataq[,unlist(lapply(dataq, is.numeric))]))
#write.csv(correlation_matrix, 'airbnbcorrelation2.csv',row.names = F)

dataq$maximum_minimum_nights = NULL;dataq$minimum_maximum_nights =
NULL;dataq$maximum_maximum_nights = NULL
dataq$number_of_reviews = NULL;dataq$availability_30 = NULL;dataq$availability_60 = NULL
dataq$availability_90 = NULL;dataq$review_scores_accuracy = NULL;dataq$host_is_superhost = NULL
dataq$is_location_exact = NULL;dataq$review_scores_communication = NULL;dataq$minimum_nights
= NULL;dataq$require_guest_phone_verification = NULL;dataq$minimum_nights_avg_ntm =
NULL;dataq$review_scores_checkin = NULL; dataq$maximum_nights =
NULL;dataq$maximum_nights_avg_ntm = NULL;dataq$host_identity_verified = NULL;
dataq$require_guest_profile_picture = NULL;dataq$host_has_profile_pic =
NULL;dataq$number_of_reviews_ltm = NULL;
dataq$reviews_per_month = NULL;dataq$instant_bookable = NULL;dataq$last_review = NULL;
dataq$first_review =NULL;dataq$host_name = NULL;dataq$license = NULL;dataq$calendar_updated =
NULL;dataq$host_verifications = NULL;dataq$monthly_price = NULL;dataq$country =
NULL;dataq$market = NULL;dataq$country_code = NULL;dataq$id = NULL;dataq$host_about =
NULL;dataq$state = NULL;dataq$jurisdiction_names = NULL;dataq$has_availability =
NULL;dataq$requires_license = NULL;dataq$is_business_travel_ready =
NULL;dataq$host_acceptance_rate = NULL;dataq$host_response_time = NULL
```

# New Variables by counting ‘amenities’ and ‘description’

#Amenities

```
dataq$amenities = as.character(dataq$amenities)
dataq$amenity_wcount = NA
dataq$amenity_wcount = as.integer(nchar(dataq$amenities))
dataq %>% select(price, amenity_wcount) %>% group_by(amenity_wcount) %>%
summary(amenity_wcount)
dataq$amenity_wcount4 = ifelse(dataq$amenity_wcount < 70,1,ifelse(dataq$amenity_wcount >69 &
dataq$amenity_wcount < 137 , 2,ifelse(dataq$amenity_wcount >136 & dataq$amenity_wcount < 173 ,
3, 4)))
dataq$amenity_wcount4 = as.integer(dataq$amenity_wcount4)
cor(dataq$price,dataq$amenity_wcount4)
cor(dataq$price,dataq$amenity_wcount)
dataq$amenity_wcount4 = NULL
```

#Description

```
dataq$description = as.character(dataq$description); dataq$descr_count = NA
dataq$descr_count = as.integer(nchar(dataq$description))
dataq %>% select(price, descr_count) %>% group_by(descr_count) %>% summary(descr_count)
dataq$descr_final = NA
dataq$descr_final = ifelse(dataq$descr_count < 1050 ,0,1)
cor(dataq$price,dataq$descr_final);dataq$descr_final = NULL;dataq$description = NULL
```

# Modifying Cleaning Fee and Security Deposit

# Cleaning Fee

```
dataq$cleaning_fee[is.na(dataq$cleaning_fee)] <- 0
c= data.frame(dataq %>% select(cleaning_fee,price) %>%
            group_by(cleaning_fee) %>% summarise(med = median(price) , n = length(cleaning_fee)))
dataq$cleaning_fee_new = NA
dataq$cleaning_fee_new = ifelse(dataq$cleaning_fee == 0, 1, ifelse(dataq$cleaning_fee > 0 &
dataq$cleaning_fee <46,2,ifelse(dataq$cleaning_fee > 45 & dataq$cleaning_fee <109,3,
ifelse(dataq$cleaning_fee > 108 & dataq$cleaning_fee <199, 4,ifelse(dataq$cleaning_fee > 198 ,5,6)))))
table(dataq$cleaning_fee_new)
dataq %>% select(cleaning_fee_new,price) %>% group_by(cleaning_fee_new) %>%
summarise(med=median(price))

dataq$cleaning_fee_new_median = NA
```

```
dataq$cleaning_fee_new_median = ifelse(dataq$cleaning_fee_new == 1, 85,
ifelse(dataq$cleaning_fee_new == 2, 72, ifelse(dataq$cleaning_fee_new == 3,
140,ifelse(dataq$cleaning_fee_new == 4, 200,300))))
table(dataq$cleaning_fee_new_median)
```

# Security Deposit

```
dataq$security_deposit[is.na(dataq$security_deposit)] <- 0
dataq$security_deposit = as.character(dataq$security_deposit)
table(dataq$neighbourhood_group_cleansed)
dataq$nb = NA
dataq$nb = dataq$neighbourhood_group_cleansed
str(dataq$security_deposit)
dataq$nb = as.character(dataq$nb)
dataq$security_deposit = ifelse(dataq$security_deposit == '0' & dataq$nb == 'Bronx','60',
ifelse(dataq$security_deposit == '0' & dataq$nb == 'Brooklyn','91',ifelse(dataq$security_deposit == '0' &
dataq$nb == 'Manhattan','145',ifelse(dataq$security_deposit == '0' & dataq$nb == 'Queens','72','70'))))
dataq$security_deposit = as.numeric(dataq$security_deposit)
dataq$nb = NULL
cor(dataq$price, dataq$security_deposit)
cor(dataq$price, dataq$cleaning_fee)
```

# BINNING

#Property Type

```
a= dataq %>% select(property_type,price) %>% group_by(property_type) %>% summarise(med =
median(price))
a = data.frame(a) ; write.csv(a, 'medianoroperty.csv',row.names = F)
dataq$property_typeold = dataq$property_type
dataq$property_type = as.character(dataq$property_type)
dataq$property_type = ifelse(dataq$property_type == 'Lighthouse'| dataq$property_type ==
'Houseboat'| dataq$property_type == 'Timeshare' | dataq$property_type == 'Resort','2',
ifelse(dataq$property_type == 'Cottage'| dataq$property_type == 'Tent'| dataq$property_type ==
'Hotel' | dataq$property_type == 'Serviced apartment' | dataq$property_type == 'Condominium' |
dataq$property_type == 'Loft'| dataq$property_type == 'Boutique hotel' | dataq$property_type ==
'Boat','2',ifelse(dataq$property_type == 'Cave'| dataq$property_type == 'Other'| dataq$property_type
== 'Apartment' | dataq$property_type == 'Nature lodge'| dataq$property_type == 'Bungalow' |
dataq$property_type == 'Guest suite'| dataq$property_type == 'Townhouse' | dataq$property_type ==
'Cabin' | dataq$property_type == 'Camper/RV','3',ifelse(dataq$property_type == 'Bed and breakfast'|
dataq$property_type == 'Aparthotel'| dataq$property_type == 'Guesthouse' | dataq$property_type ==
'House'| dataq$property_type == 'Hostel' | dataq$property_type == 'Loft'| dataq$property_type ==
'Boutique hotel' | dataq$property_type == 'Earth house','4',ifelse(dataq$property_type == 'Tiny
house'|dataq$property_type == 'Dome house'| dataq$property_type == 'Villa' |dataq$property_type
== 'Pension (South Korea)','4','5')))))
dataq$property_type = as.factor(dataq$property_type)
```

```
table(dataq$property_type)
```

# neighbourhood_cleansed

```
b= dataq %>% select(neighbourhood_cleansed,price) %>% group_by(neighbourhood_cleansed) %>%
summarise(medianprice = median(price) , n = length(neighbourhood_cleansed)); b= data.frame(b)
write.csv(b, 'neighbour.csv',row.names = F)
dataq$neighbourhood_before = NA
dataq$neighbourhood_before = dataq$neighbourhood_group_cleansed
dataq$neighbourhood_cleansed = as.character(dataq$neighbourhood_cleansed)

dataq$neighbourhood_cleansed = ifelse(dataq$neighbourhood_cleansed ==
'Neponsit'|dataq$neighbourhood_cleansed == 'Tribeca'| dataq$neighbourhood_cleansed ==
'Willowbrook','1',ifelse(dataq$neighbourhood_cleansed ==
'NoHo','2',ifelse(dataq$neighbourhood_cleansed == 'Flatiron
District','3',ifelse(dataq$neighbourhood_cleansed == 'Chelsea'|
dataq$neighbourhood_cleansed == 'West Village'| dataq$neighbourhood_cleansed == 'SoHo','4',
ifelse(dataq$neighbourhood_cleansed == 'Theater District'|dataq$neighbourhood_cleansed == 'Breezy
Point'|dataq$neighbourhood_cleansed == 'DUMBO','5',ifelse(dataq$neighbourhood_cleansed ==
'Midtown','6',  ifelse(dataq$neighbourhood_cleansed == 'Greenwich
Village'|dataq$neighbourhood_cleansed == 'Tottenville'|  dataq$neighbourhood_cleansed == 'Financial
District' | dataq$neighbourhood_cleansed == 'Nolita', '7',
ifelse(dataq$neighbourhood_cleansed == 'Murray Hill'|dataq$neighbourhood_cleansed == 'Battery Park
City'|dataq$neighbourhood_cleansed == 'Belle Harbor','8',ifelse(dataq$neighbourhood_cleansed ==
'Holliswood'|dataq$neighbourhood_cleansed == 'Gramercy'| dataq$neighbourhood_cleansed ==
'Brooklyn Heights' |  dataq$neighbourhood_cleansed == 'Sea Gate' | dataq$neighbourhood_cleansed ==
'Kips Bay' | dataq$neighbourhood_cleansed == 'Lighthouse Hill','9',
ifelse(dataq$neighbourhood_cleansed == "Hell's Kitchen"| dataq$neighbourhood_cleansed == 'East
Village'| dataq$neighbourhood_cleansed == 'Upper West Side' | dataq$neighbourhood_cleansed ==
'Carroll Gardens' | dataq$neighbourhood_cleansed == 'Stuyvesant Town' |
dataq$neighbourhood_cleansed == 'Navy Yard' | dataq$neighbourhood_cleansed == 'Riverdale' |
dataq$neighbourhood_cleansed == 'Lower East Side' , '10' ,  ifelse(dataq$neighbourhood_cleansed ==
'Civic Center' , '11' ,ifelse(dataq$neighbourhood_cleansed ==
'Vinegar Hill'| dataq$neighbourhood_cleansed == 'Upper East Side'| dataq$neighbourhood_cleansed ==
'Park Slope' | dataq$neighbourhood_cleansed == 'Boerum Hill' | dataq$neighbourhood_cleansed ==
'Little Italy' | dataq$neighbourhood_cleansed == 'Cobble Hill' | dataq$neighbourhood_cleansed ==
'Bay Terrace' , '12' ,ifelse(dataq$neighbourhood_cleansed == 'South Slope'|
dataq$neighbourhood_cleansed ==    'Downtown Brooklyn'| dataq$neighbourhood_cleansed ==
'Chinatown' | dataq$neighbourhood_cleansed ==  'Columbia St' | dataq$neighbourhood_cleansed ==
'Fort Greene' | dataq$neighbourhood_cleansed == 'Gowanus' | dataq$neighbourhood_cleansed ==
'Huguenot' | dataq$neighbourhood_cleansed == 'Manhattan Beach' , '13' ,
ifelse(dataq$neighbourhood_cleansed == 'Clinton Hill' | dataq$neighbourhood_cleansed == 'Windsor
Terrace' ,'14', ifelse(dataq$neighbourhood_cleansed == 'Prospect Heights'|
dataq$neighbourhood_cleansed == 'Gerritsen Beach'|
dataq$neighbourhood_cleansed == 'Greenpoint' | dataq$neighbourhood_cleansed == "Prince's Bay" |
dataq$neighbourhood_cleansed == 'Grymes Hill' , '15' ,ifelse(dataq$neighbourhood_cleansed ==
'Williamsburg'|  dataq$neighbourhood_cleansed == 'Two Bridges' | dataq$neighbourhood_cleansed ==
'Rosebank' | dataq$neighbourhood_cleansed == 'Grymes Hill' , '16'
,ifelse(dataq$neighbourhood_cleansed == 'East Harlem'|
dataq$neighbourhood_cleansed == 'Red Hook'| dataq$neighbourhood_cleansed == 'Arverne' |
dataq$neighbourhood_cleansed == 'Unionport' | dataq$neighbourhood_cleansed == 'Spuyten Duyvil' |
dataq$neighbourhood_cleansed == 'Morningside Heights' | dataq$neighbourhood_cleansed ==
'Rockaway Beach' |dataq$neighbourhood_cleansed == 'Middle Village' |
dataq$neighbourhood_cleansed == 'Great Kills' |
dataq$neighbourhood_cleansed == 'Todt Hill' | dataq$neighbourhood_cleansed == 'Glen Oaks' |
dataq$neighbourhood_cleansed == 'Bergen Beach' | dataq$neighbourhood_cleansed == 'City Island' ,
'17',  ifelse(dataq$neighbourhood_cleansed == 'Howard Beach' | dataq$neighbourhood_cleansed ==
'Harlem' | dataq$neighbourhood_cleansed == 'Long Island City' | dataq$neighbourhood_cleansed ==
'Mariners Harbor' | dataq$neighbourhood_cleansed == 'Whitestone' | dataq$neighbourhood_cleansed
== 'Ditmars Steinway' | dataq$neighbourhood_cleansed == 'Forest Hills' , '18'
,ifelse(dataq$neighbourhood_cleansed == 'Hollis'|
dataq$neighbourhood_cleansed == 'Kew Gardens Hills'| dataq$neighbourhood_cleansed == 'Crown
Heights' | dataq$neighbourhood_cleansed == 'Astoria' | dataq$neighbourhood_cleansed == 'Prospect-
Lefferts Gardens' |  dataq$neighbourhood_cleansed == 'Roosevelt Island' |
dataq$neighbourhood_cleansed == 'Ozone Park' |
dataq$neighbourhood_cleansed == 'Pelham Bay' | dataq$neighbourhood_cleansed == 'Shore Acres' |
dataq$neighbourhood_cleansed == 'Midland Beach' | dataq$neighbourhood_cleansed == 'Mill Basin' ,
'19',  ifelse(dataq$neighbourhood_cleansed == 'Throgs Neck'| dataq$neighbourhood_cleansed == 'Van
Nest'|dataq$neighbourhood_cleansed == 'Bedford-Stuyvesant' | dataq$neighbourhood_cleansed ==
'Bay Terrace, Staten Island'   | dataq$neighbourhood_cleansed == 'East New York' |
dataq$neighbourhood_cleansed == 'Fort Hamilton' |
dataq$neighbourhood_cleansed == 'Richmondtown' | dataq$neighbourhood_cleansed == 'Dongan Hills'
|  dataq$neighbourhood_cleansed == 'Sunset Park' | dataq$neighbourhood_cleansed == 'Bay Ridge' ,
'20' , ifelse(dataq$neighbourhood_cleansed == 'Brighton Beach'| dataq$neighbourhood_cleansed ==
'Marble Hill'|  dataq$neighbourhood_cleansed == 'Canarsie' | dataq$neighbourhood_cleansed ==
'Bayside' | dataq$neighbourhood_cleansed == 'Bayswater' | dataq$neighbourhood_cleansed == 'East
Flatbush' | dataq$neighbourhood_cleansed == 'Springfield Gardens' | dataq$neighbourhood_cleansed
== 'Glendale' |dataq$neighbourhood_cleansed == 'Arrochar' | dataq$neighbourhood_cleansed == 'West
Brighton' |dataq$neighbourhood_cleansed == 'Edenwald' | dataq$neighbourhood_cleansed ==
'Baychester' |dataq$neighbourhood_cleansed == 'Co-op City' | dataq$neighbourhood_cleansed ==
'Rossville' |dataq$neighbourhood_cleansed == 'Mott Haven' | dataq$neighbourhood_cleansed ==
'Kingsbridge' , '21' , ifelse(dataq$neighbourhood_cleansed == 'Queens Village' |
dataq$neighbourhood_cleansed == 'Sunnyside' |
dataq$neighbourhood_cleansed == 'South Ozone Park' | dataq$neighbourhood_cleansed == 'Grant City'
| dataq$neighbourhood_cleansed == 'Concourse' | dataq$neighbourhood_cleansed == 'St. George' |
dataq$neighbourhood_cleansed == 'Graniteville' , '22' ,ifelse(dataq$neighbourhood_cleansed ==
'Gravesend'|dataq$neighbourhood_cleansed == 'Washington Heights'| dataq$neighbourhood_cleansed
== 'Flatbush' | dataq$neighbourhood_cleansed == 'Kensington' | dataq$neighbourhood_cleansed ==
'Maspeth' |  dataq$neighbourhood_cleansed == 'Bensonhurst' | dataq$neighbourhood_cleansed ==
'Pelham Gardens' | dataq$neighbourhood_cleansed == 'Bath Beach' | dataq$neighbourhood_cleansed
== 'Clifton' |dataq$neighbourhood_cleansed == 'Jackson Heights' | dataq$neighbourhood_cleansed ==
'Fresh Meadows' |dataq$neighbourhood_cleansed == 'Dyker Heights' | dataq$neighbourhood_cleansed
== 'Randall Manor' |  dataq$neighbourhood_cleansed == 'Midwood' | dataq$neighbourhood_cleansed
== 'Westchester Square' |dataq$neighbourhood_cleansed == 'Coney Island' |
dataq$neighbourhood_cleansed == 'Cypress Hills' |
dataq$neighbourhood_cleansed == 'South Beach' | dataq$neighbourhood_cleansed == 'Eltingville' |
dataq$neighbourhood_cleansed == 'Flatlands' | dataq$neighbourhood_cleansed == 'Tompkinsville' |
dataq$neighbourhood_cleansed == 'West Farms ', '23' ,ifelse(dataq$neighbourhood_cleansed ==
'Bushwick'|dataq$neighbourhood_cleansed == 'Castle Hill'| dataq$neighbourhood_cleansed ==
'Ridgewood' |
dataq$neighbourhood_cleansed == 'Jamaica' | dataq$neighbourhood_cleansed == 'Woodside' |
dataq$neighbourhood_cleansed == 'Jamaica Estates' | dataq$neighbourhood_cleansed == 'Laurelton' |
dataq$neighbourhood_cleansed == 'Morris Park' | dataq$neighbourhood_cleansed == 'East Morrisania'
|dataq$neighbourhood_cleansed == 'Douglaston' | dataq$neighbourhood_cleansed == 'Bellerose' |
dataq$neighbourhood_cleansed == 'Jamaica Hills' | dataq$neighbourhood_cleansed == 'Oakwood', '24' ,
ifelse(dataq$neighbourhood_cleansed == 'Rosedale'| dataq$neighbourhood_cleansed == 'Rego Park'|
dataq$neighbourhood_cleansed == 'Flushing' | dataq$neighbourhood_cleansed == 'Sheepshead Bay' |
dataq$neighbourhood_cleansed == 'Richmond Hill' | dataq$neighbourhood_cleansed == 'Brownsville' |
dataq$neighbourhood_cleansed == 'Longwood' | dataq$neighbourhood_cleansed == 'Fordham' |
dataq$neighbourhood_cleansed == 'Williamsbridge' | dataq$neighbourhood_cleansed == 'Kew Gardens'
|  dataq$neighbourhood_cleansed == 'Wakefield' | dataq$neighbourhood_cleansed == 'Concourse
Village' | dataq$neighbourhood_cleansed == 'College Point' | dataq$neighbourhood_cleansed ==
'Eastchester' |
dataq$neighbourhood_cleansed == 'Fieldston' | dataq$neighbourhood_cleansed == 'Morrisania' |
dataq$neighbourhood_cleansed == 'New Springville' , '25' ,ifelse(dataq$neighbourhood_cleansed ==
'Elmhurst'|
dataq$neighbourhood_cleansed == 'St. Albans'| dataq$neighbourhood_cleansed == 'Briarwood' |
dataq$neighbourhood_cleansed == 'Cambria Heights' | dataq$neighbourhood_cleansed ==
'Woodlawn'|   dataq$neighbourhood_cleansed == 'Bronxdale' | dataq$neighbourhood_cleansed ==
'Claremont Village'|
dataq$neighbourhood_cleansed == 'Morris Heights' | dataq$neighbourhood_cleansed == 'Edgemere' |
dataq$neighbourhood_cleansed == 'North Riverdale' | dataq$neighbourhood_cleansed == 'Castleton
Corners' , '26' ,
ifelse(dataq$neighbourhood_cleansed == 'Norwood'| dataq$neighbourhood_cleansed == 'Silver Lake'|
dataq$neighbourhood_cleansed == 'East Elmhurst' | dataq$neighbourhood_cleansed == 'Stapleton' |
dataq$neighbourhood_cleansed == 'Arden Heights' | dataq$neighbourhood_cleansed == 'Port Morris' |
dataq$neighbourhood_cleansed == 'Allerton' | dataq$neighbourhood_cleansed == 'University Heights' |
dataq$neighbourhood_cleansed == 'Mount Hope' | dataq$neighbourhood_cleansed == 'New Brighton' |
dataq$neighbourhood_cleansed == 'Olinville' , '27' ,ifelse(dataq$neighbourhood_cleansed == 'Borough
Park'|dataq$neighbourhood_cleansed == 'Parkchester'| dataq$neighbourhood_cleansed ==
'Highbridge' |
dataq$neighbourhood_cleansed == 'Woodhaven' | dataq$neighbourhood_cleansed == 'Far Rockaway' |
dataq$neighbourhood_cleansed == 'Melrose' | dataq$neighbourhood_cleansed == 'Emerson Hill' |
dataq$neighbourhood_cleansed == 'Clason Point' | dataq$neighbourhood_cleansed == 'Belmont' |
dataq$neighbourhood_cleansed == 'Little Neck' | dataq$neighbourhood_cleansed == 'Soundview' |
dataq$neighbourhood_cleansed == 'Pleasant Plains', '28' ,ifelse(dataq$neighbourhood_cleansed ==
'Corona'|
dataq$neighbourhood_cleansed == 'Port Richmond'| dataq$neighbourhood_cleansed == 'Mount Eden'|
dataq$neighbourhood_cleansed == 'Hunts Point' | dataq$neighbourhood_cleansed == 'Tremont' |
dataq$neighbourhood_cleansed == 'Westerleigh' | dataq$neighbourhood_cleansed == 'Schuylerville' |
dataq$neighbourhood_cleansed == 'Concord' , '29' ,
ifelse(dataq$neighbourhood_cleansed == "Bull's Head" ,'30', '31' ))))))))))))))))))))))))))))))
dataq$neighbourhood_cleansed = as.factor(dataq$neighbourhood_cleansed)
dataq$neighbourhood_group_cleansed = as.numeric(dataq$neighbourhood_group_cleansed)
```

#neighbour variable - median price binned

```
dataq$neighbourhood_cleansed = as.character(dataq$neighbourhood_cleansed)
dataq$neighbourhood_cleansed_num = NA

dataq$neighbourhood_cleansed_num = ifelse(dataq$neighbourhood_cleansed == '1','259',
ifelse(dataq$neighbourhood_cleansed == '2','243',ifelse(dataq$neighbourhood_cleansed ==
'3','221',ifelse(dataq$neighbourhood_cleansed == '4','199',ifelse(dataq$neighbourhood_cleansed ==
'5','195',ifelse(dataq$neighbourhood_cleansed == '6','189',ifelse(dataq$neighbourhood_cleansed ==
'7','180',ifelse(dataq$neighbourhood_cleansed == '8','175',ifelse(dataq$neighbourhood_cleansed ==
'9','160',ifelse(dataq$neighbourhood_cleansed == '10','150',ifelse(dataq$neighbourhood_cleansed ==
'11','145',ifelse(dataq$neighbourhood_cleansed == '12','140',ifelse(dataq$neighbourhood_cleansed ==
'13','130',ifelse(dataq$neighbourhood_cleansed == '14','125',ifelse(dataq$neighbourhood_cleansed ==
'15','120',ifelse(dataq$neighbourhood_cleansed == '16','110',ifelse(dataq$neighbourhood_cleansed ==
'17','99',ifelse(dataq$neighbourhood_cleansed == '18','90',ifelse(dataq$neighbourhood_cleansed ==
'19','85',ifelse(dataq$neighbourhood_cleansed == '20','80',ifelse(dataq$neighbourhood_cleansed ==
'21','75',ifelse(dataq$neighbourhood_cleansed == '22','72',ifelse(dataq$neighbourhood_cleansed ==
'23','70',ifelse(dataq$neighbourhood_cleansed == '24','65',ifelse(dataq$neighbourhood_cleansed ==
'25','60',ifelse(dataq$neighbourhood_cleansed == '26','59',ifelse(dataq$neighbourhood_cleansed ==
'27','55',ifelse(dataq$neighbourhood_cleansed == '28','50',ifelse(dataq$neighbourhood_cleansed ==
'29','40',ifelse(dataq$neighbourhood_cleansed == '30','25'
,ifelse(dataq$neighbourhood_cleansed == '31','70','1')))))))))))))))))))))))))))))))

dataq$neighbourhood_cleansed_num = as.numeric(dataq$neighbourhood_cleansed_num)
dataq$neighbourhood_cleansed = as.factor(dataq$neighbourhood_cleansed)
```

#bedrooms variable - median price binned

```
dataq %>% select(bedrooms,price) %>% group_by(bedrooms) %>% summarise(med = median(price))
dataq$bedrooms = as.character(dataq$bedrooms)
dataq$bedrooms_num = NA
dataq$bedrooms_num = ifelse(dataq$bedrooms == '0','129',ifelse(dataq$bedrooms ==
'1','89',ifelse(dataq$bedrooms == '2','180',ifelse(dataq$bedrooms == '3','250',ifelse(dataq$bedrooms ==
'4','322',ifelse(dataq$bedrooms == '5','399',ifelse(dataq$bedrooms == '6','562',ifelse(dataq$bedrooms
== '7','572',ifelse(dataq$bedrooms == '8','500',ifelse(dataq$bedrooms == '9','500'
,ifelse(dataq$bedrooms == '10','695','135'))))))))))

dataq$bedrooms_num= as.numeric(dataq$bedrooms_num)
dataq$bedrooms = as.numeric(dataq$bedrooms)
```

#bathrooms variable - median price binned

```
dataq %>% select(bathrooms,price) %>% group_by(bathrooms) %>% summarise(med = median(price))
dataq$bathrooms = as.character(dataq$bathrooms)
dataq$bathrooms_num = NA
dataq$bathrooms_num = ifelse(dataq$bathrooms == '0','85',ifelse(dataq$bathrooms ==
'0.5','75',ifelse(dataq$bathrooms == '1','100',ifelse(dataq$bathrooms ==
'1.5','88.5',ifelse(dataq$bathrooms == '2','175',ifelse(dataq$bathrooms ==
'2.5','250',ifelse(dataq$bathrooms == '3','154',ifelse(dataq$bathrooms ==
'3.5','414',ifelse(dataq$bathrooms == '4','75',ifelse(dataq$bathrooms ==
'4.5','694',ifelse(dataq$bathrooms == '5','760',ifelse(dataq$bathrooms ==
'5.5','448',ifelse(dataq$bathrooms == '6','40',ifelse(dataq$bathrooms ==
'6.5','35',ifelse(dataq$bathrooms == '7','50',
ifelse(dataq$bathrooms == '8','55','135')))))))))))))))

dataq$bathrooms_num= as.numeric(dataq$bathrooms_num)
dataq$bathrooms = as.numeric(dataq$bathrooms)
```

#bed_type variable - median price binned

```
dataq %>% select(bed_type,price) %>% group_by(bed_type) %>% summarise(med = median(price))
dataq$bed_type = as.character(dataq$bed_type)
dataq$bed_type_num = NA
dataq$bed_type_num = ifelse(dataq$bed_type == 'Airbed','79',ifelse(dataq$bed_type ==
'Couch','55',ifelse(dataq$bed_type == 'Futon','75',ifelse(dataq$bed_type == 'Pull-out
Sofa','90',ifelse(dataq$bed_type == 'Real Bed','105','136')))))
dataq$bed_type_num= as.numeric(dataq$bed_type_num)
dataq$bed_type = as.factor(dataq$bed_type)
```

#extra people variable - median price binned

```
extrapeopledata = data.frame(dataq %>% select(extra_people,price) %>% group_by(extra_people) %>%
summarise(med = length(price)))
write.csv(extrapeopledata,'extra2.csv')
dataq$extra_people_binned = NA
dataq$extra_people_binned = ifelse(dataq$extra_people == 0 ,1,ifelse(dataq$extra_people >0 &
dataq$extra_people <25 ,2,ifelse(dataq$extra_people >24 & dataq$extra_people <100 ,3,4)))
dataq$extra_people_binned = as.factor(dataq$extra_people_binned)
```

#property_type variable - median price binned

```
table(dataq$property_type)
```

```
dataq %>% select(property_type,price) %>% group_by(property_type) %>% summarise(med =
median(price))
dataq$property_type_num = NA
dataq$property_type_num = as.character(dataq$property_type)
dataq$property_type_num = ifelse(dataq$property_type == '2' ,'151',ifelse(dataq$property_type == '3'
,'105',ifelse(dataq$property_type == '4' ,'70','135')))
dataq$property_type = as.factor(dataq$property_type)
dataq$property_type_num = as.numeric(dataq$property_type_num)
```

# host_total_listings_count

```
dataq$host_list_final = NA
dataq$host_list_final = dataq$host_total_listings_count
dataq$host_list_final = as.character(dataq$host_list_final)
dataq$host_list_final = ifelse(dataq$host_total_listings_count < 6,1,
ifelse(dataq$host_total_listings_count  >5 & dataq$host_total_listings_count  < 250 , 2,
ifelse(dataq$host_total_listings_count  >249 & dataq$host_total_listings_count<500, 3,
ifelse(dataq$host_total_listings_count  >499 & dataq$host_total_listings_count <1000,4,5))))
dataq$host_list_final = as.factor(dataq$host_list_final)
```

# minimum minimum nights

```
str(dataq$minimum_minimum_nights)
dataq %>% select(price, minimum_minimum_nights) %>% group_by(minimum_minimum_nights) %>%
summary(minimum_minimum_nights)
dataq$min_night = NA
dataq$min_night = ifelse(dataq$minimum_minimum_nights < 70,1,
ifelse(dataq$minimum_minimum_nights  >69 & dataq$minimum_minimum_nights  < 105 , 2,
ifelse(dataq$minimum_minimum_nights  >104 & dataq$minimum_minimum_nights  <173, 3, 4)))

cor(dataq$price, dataq$min)
dataq$minimum_minimum_nights = NULL
```

Text Mining (Amenities)

```
dataq$amenities = as.character(dataq$amenities)
corpus_review=Corpus(VectorSource(dataq$amenities))
corpus_review=tm_map(corpus_review, tolower)
corpus_review=tm_map(corpus_review, removePunctuation)
corpus_review=tm_map(corpus_review, removeWords, stopwords("english"))
corpus_review=tm_map(corpus_review, stemDocument)
term_count <- freq_terms(corpus_review, 20)

# dtm & tdm
review_dtm <- DocumentTermMatrix(corpus_review)
```

```
review_tdm <- TermDocumentMatrix(corpus_review)
review_m <- as.matrix(review_tdm)
#glimpse(review_m); dim(review_m)
review_term_freq <- rowSums(review_m)
review_term_freq <- sort(review_term_freq, decreasing = T)
review_term_freq[1:10]

##Create bi-grams
review_bigram <- tokens(dataq$amenities) %>%
  tokens_remove("\\p{P}", valuetype = "regex", padding = TRUE) %>%
  tokens_remove(stopwords("english"), padding  = TRUE) %>%
  tokens_ngrams(n = 2) %>%
  dfm()
```

###Tokenization (Amenity)

```
# Tokenize descriptions
reviewtokens=tokens(dataq$amenities,what="word",remove_numbers=TRUE,remove_punct=TRUE,
remove_symbols=TRUE, remove_hyphens=TRUE)
# Lowercase the tokens
reviewtokens=tokens_tolower(reviewtokens)
# remove stop words and unnecessary words
rmwords <- c("dress", "etc", "also", "xxs", "xs", "s")
reviewtokens=tokens_select(reviewtokens, stopwords(),selection = "remove")
reviewtokens=tokens_remove(reviewtokens,rmwords)
# Stemming tokens
reviewtokens=tokens_wordstem(reviewtokens,language = "english")
reviewtokens=tokens_ngrams(reviewtokens,n=1:6)
# Creating a bag of words
reviewtokensdfm=dfm(reviewtokens,tolower = FALSE)
# Remove sparsity
reviewSparse <- convert(reviewtokensdfm, "tm")
tm::removeSparseTerms(reviewSparse, 0.7)
# Create the dfm
dfm_trim(reviewtokensdfm, min_docfreq = 0.3)
x=dfm_trim(reviewtokensdfm, sparsity = 0.98)
## Setup a dataframe with features
df=convert(x,to="data.frame")
df2  = cbind(dataq,df)
correlation_matrix = data.frame(cor(df2[,unlist(lapply(df2, is.numeric))]))
write.csv(correlation_matrix, 'airbnbcorrelation3.csv',row.names = F)

positives = c('iron_laptop_friend', 'elev', 'tv_wifi_air',  'gym',
'gym_elev','play_travel','tv','friend_washer_dryer','dryer','friend_washer',
        'kid_friend_washer','friend','heat_famili','heat_famili_kid',
        'washer_dryer','famili','kid','famili_kid','kid_friend',

'famili_kid_friend','washer_dryer_smoke','dryer_smoke','dryer_smoke_detector','washer','maker_refriger_dishwash','dryer_iron_laptop','tv_cabl','tv_cabl_tv','dishwash','cabl','cabl_tv','refriger_dishwash','refriger_dishwash_dish','dishwash_dish','dishwash_dish_silverwar')

positivedf <- df[,colnames(df) %in% positives]
positivedf$total = rowSums(positivedf, na.rm=T)

negatives =
c('heat_smoke','heat_smoke_detector','bedroom','lock_bedroom','bedroom_door','calculated_host_listings_count_private_rooms','wifi_kitchen','lock','door','essenti_lock','essenti_lock_bedroom','miss_translat','translat_miss_translat',
'miss_translat_miss','translat','miss','translat_miss','park_heat_smoke','door_hanger','bedroom_door_hanger',
'weekly_price','kitchen_free','calculated_host_listings_count_shared_rooms','park_heat','street_park_heat')

negativedf <- df[,colnames(df) %in% negatives]
negativedf[negativedf==1]=-1
negativedf[negativedf==2]=-1
negativedf[negativedf==3]=-1
negativedf$totalneg = rowSums(negativedf, na.rm=T)

consolidated_df = cbind(positivedf,negativedf)
consolidated_df$finalsum = positivedf$total + negativedf$totalneg
consolidated_df$finalsum_bin = NA
consolidated_df$finalsum_bin = as.character(consolidated_df$finalsum)
consolidated_df$finalsum_bin  = ifelse(consolidated_df$finalsum_bin < -10,1,
ifelse(consolidated_df$finalsum_bin >-11 & consolidated_df$finalsum_bin < 0 , 2,
ifelse(consolidated_df$finalsum_bin >-1 & consolidated_df$finalsum_bin <20,3,4)))

consolidated_df$finalsum_bin = as.factor(consolidated_df$finalsum_bin)

finalcolumns = c('finalsum_bin','finalsum','elev','dryer_iron_laptop',
          'washer','dryer','dryer_smoke','gym_elev','tv','kitchen_free','park_heat_smoke','gym'
          ,'play_travel','dryer_iron_laptop','dishwash_dish','refriger_dishwash')

consolidated_df <- consolidated_df[,colnames(consolidated_df) %in% finalcolumns]
```

#Final Adjustments Before Prediction (Creating Derivative Variables)

```
datap = cbind(consolidated_df,dataq)
datap$review_multiplied = NA
```

```
datap$review_multiplied = datap$review_scores_cleanliness * datap$review_scores_location * datap$review_scores_rating
```

# Creating Derivative Variables

```
datap$room_type = as.character(datap$room_type)
datap$room_type_numeric = ifelse(datap$room_type == 'Entire home/apt',155,ifelse(datap$room_type == 'Private room',70, ifelse(datap$room_type == 'Shared room',45,0)))
datap$room_type = as.factor(datap$room_type)

datap$highpredictors = NA
datap$room_type_numeric = as.numeric(datap$room_type_numeric)
datap$highpredictors =  datap$bedrooms_num * datap$bathrooms * datap$neighbourhood_cleansed_num * datap$room_type_numeric * datap$accommo_binned * datap$cleaning_fee_new_median * datap$review_multiplied
cor(datap$highpredictors,datap$price)

datap$bed_bath_numerics =  datap$property_type_num * datap$bed_type_num * datap$bathrooms_num * datap$bathrooms_num
cor(datap$bed_bath_numerics,datap$price)

datap$highpred2 = datap$highpredictors * datap$accommo_binned * datap$neighbourhood_cleansed_num * datap$neighbourhood_cleansed_num * datap$neighbourhood_cleansed_num
```

# Final removals based on decision tree performance

```
datap$review_scores_cleanliness = NULL
datap$calculated_host_listings_count_private_rooms = NULL
datap$weekly_price = NULL
datap$room_type_numeric = NULL
datap$descr_count = NULL
datap$host_response_rate = NULL
datap$amenity_wcount4 = NULL
datap$calculated_host_listings_count_entire_homes = NULL
datap$host_listings_count = NULL
datap$amenities = NULL
datap$access = NULL
datap$gym_elev = NULL
```

#PREDICTION

```
library(caret):library(rpart):library(rpart.plot)
```

```
datap = na.omit(datap)
set.seed(1031)
split = createDataPartition(datap$price,p = 0.7, list = F)
train = datap[split,]
test = datap[-split,]

## Decision Tree ##
tree=rpart(price ~ ., data = train, method="anova", control = rpart.control(minsplit = 200,
minbucket = 30, cp = 0.0001))
printcp(tree); plotcp(tree)
##Prune down the tree
bestcp=tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]
ptree=prune(tree,cp=bestcp)
rpart.plot(ptree,cex = 0.6)
prp(ptree, faclen = 0, cex = 0.5, extra = 2)
pred = predict(ptree)
rmse = sqrt(mean((pred-train$price)^2)); rmse
pred2 = predict(ptree,newdata = test)
rmse2 = sqrt(mean((pred2-test$price)^2)); rmse2

## linear model ##
lm = lm(price~.,data=train)
pred = predict(lm)
rmse = sqrt(mean((pred-train$price)^2)); rmse
pred2 = predict(lm,newdata = test)
rmse2 = sqrt(mean((pred2-test$price)^2)); rmse2
summary(lm)

 ## rf model ##
library(randomForest)
set.seed(617)
forest = randomForest(price~.,data=train,ntree = 700)
pred = predict(forest)
rmse = sqrt(mean((pred-train$price)^2));rmse
pred = predict(forest)
rmse = sqrt(mean((pred-test$price)^2));rmse
importance(forest) # relative importance of predictors (highest <-> most important)
varImpPlot(forest) # plot results

## boosting model ##
library(gbm)
set.seed(617)
```

```
boosted_model = gbm(price~.,data=datap,shrinkage = 0.01,     interaction.depth = 3,n.minobsinnode = 5, n.trees = 5000,cv.folds = 7)
pred = predict(boosted_model)
rmse = sqrt(mean((pred-train$price)^2)); rmse
pred2 = predict(boosted_model,newdata = test)
rmse2 = sqrt(mean((pred2-test$price)^2)); rmse2
```

# Read scoring data and apply model to generate predictions

```
setwd('C:/Users/USER01/Desktop/R/Kaggle')
scoringData = read.csv('scoringData.csv')
dataq = scoringData
```

COPY PASTE EXACT THE SAME CODE ABOVE APPLIED FOR THE ANALYSIS DATA

# Construct submission from predictions

```
pred = predict(forest,newdata=datap)
submissionFile = data.frame(id = datap$id, price = forest)
write.csv(submissionFile, 'kerim_submission_forest.csv',row.names = F)
```