Analysis Of Airbnb Listings

Sai Venkata Manoj Vungarala | Ishal Abhishek Mummidivarapu | Sujith Naarayan Hirendra Babu

Summary

The goal of the project is to analyze the Airbnb listings in six major cities (New York, Los Angeles, San Francisco, Boston, Chicago, and DC) in the United States. Airbnb Inc. is an American company that operates an online marketplace for lodging, primarily homestays for vacation rentals, and tourism activities. It lists more than 6 million rooms and houses in 81,000 cities. The platform has more than 150 million users worldwide with 2 million people staying in Airbnb rentals across the world on any given night. Given the extremely large quantity of listings, this project's dataset will contain listing in only six US cities (New York, Los Angeles, San Francisco, Boston, Chicago, and DC).

The data contains 74111 rows and 29 columns. Some important columns are price, property_type, room_type, bedrooms, bathrooms, city, amenities, etc.. Apart from analysis, this project will also include modeling and will list out the important factors while listing a property on the website. Additionally, the project also provides the basic common amenities required while listing a property by utilizing the concept of tokenization.

A brief description of the project's methods would include:

1. Data Preprocessing: Tidying the data
2. Data Visualization : Visualizing various graphs and making conclusions based on that.
3. Tokenization : Helps in identifying the top 10 common amenities in a listing.
4. Modeling : Applied Linear Regression, Random Forest and SVM which will list out the best response variable for price predictor.

In the results sections, the project will conclude the important factors like amenities, type of the listing, etc that are to be considered while listing an Airbnb property.

Methods

Data Preprocessing:

The dataset was taken from Kaggle (link) with the title "US Airbnb Listings". Our Airbnb dataset consists of 29 columns that describe the type of the property, price, amenities, number of rooms, number of bathrooms, etc.

The first thing that was performed was to add the state column so that it would be easier to plot maps. Next, the log_price column has been converted to regular price so that would be easier to understand. Finally, tokenization has been performed to amenities column to identify the top ten common amenities provided by a listing.

Data Visualization:

Exploratory Data Analysis (*EDA*) helps us in analyzing the data sets to visually summarize their characteristics. The various data from the dataset are visualized using various graphs to provide greater insights about the dataset. Below are the visualizations that this project consists of:

- Number of Airbnb's in Different Cities
- Property Type and its Count
- Count of Room Types
- Average price of Property Type
- Top 10 Amenities
- No. of Airbnb's with Cancellation policy in the six Cities
- Instant Booking option
- Average Ratings in different cities
- Number of Airbnb's Vs type of accommodations in various cities
- Geological location of the listings and its Price Range

Tokenization:

The amenities column consists of a list of amenities that a particular listing offers. Tokenization (process of splitting text into tokens) is used to read and identify that top 10 common amenities that most of the listings offered.

Modeling:

1 Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

While training the model we are given: x: input training data (univariate – one input variable(parameter)) y: labels to data (supervised learning) When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best $\theta 1$ and $\theta 2$ values. $\theta 1$: intercept $\theta 2$: coefficient of x Once we find the best $\theta 1$ and $\theta 2$ values, we get the best fit line. So, when we are finally using our model for prediction, it will predict the value of y for the input value of x. How to update $\theta 1$ and $\theta 2$ values to get the best fit line? Cost Function (J): By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the $\theta 1$ and $\theta 2$ values, to reach the best value that minimizes the error between predicted y value (pred) and true y value (y).

2 Support Vector Machines (SVM):

A support vector machine (SVM) is a linear classifier that finds a hyperplane in an n-dimensional space, where n is the total number of features. There are many different planes to draw to separate the features, but the goal of SVM is to identify one that has the greatest distance between the vectors of both classes. These hyperplanes serve as judgment boundaries, allowing the data to be classified more precisely. Kernels are used to transform data, and we picked the linear kernel because our data is linearly separable into two classes. The data was vectorized with the help of the TF-IDF algorithm.

3 Random Forest:

The RF classifier is an ensemble method for bagging that involves training several decision trees in parallel with bootstrapping and aggregation. Using different subsets of available features, several individual decision trees are trained in parallel on different subsets of the training dataset. Bootstrapping ensures that each decision tree in the random forest is unique, lowering the RF classifier's overall variance. RF classifier aggregates individual tree decisions for the final decision; as a result, RF classifier has good generalization. In terms of accuracy, the RF classifier tends to outperform most other classification methods without the risk of overfitting. RF classifiers, like DT classifiers, do not require feature scaling. RF classifier, unlike DT classifier, is more resistant to training sample selection and noise in the training dataset. In comparison to the DT classifier, the RF classifier is more difficult to interpret but easier to tune the hyperparameter.

Results

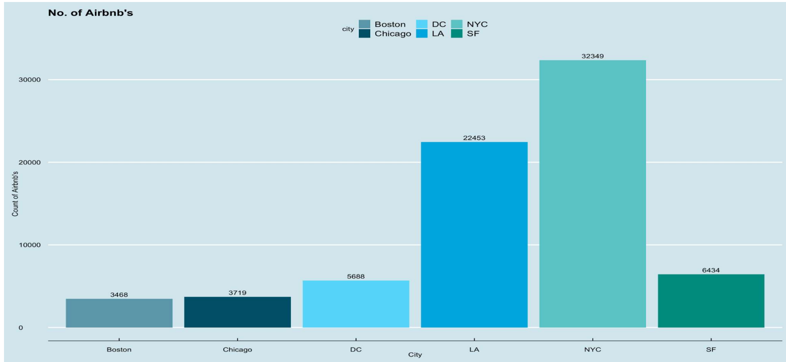


Figure 1. Number of Airbnb's in Different Cities

Figure 1. Shows the graph plot between Count of Airbnb's in the six Cities. Based on the Airbnb listings of major US cities it is observed that New York City (32349) being the financial capital has the maximum number of listings followed by Los Angeles (22453) and the least being Boston (3468). From the graph, we can also infer that the city having many Airbnb's is densely populated.

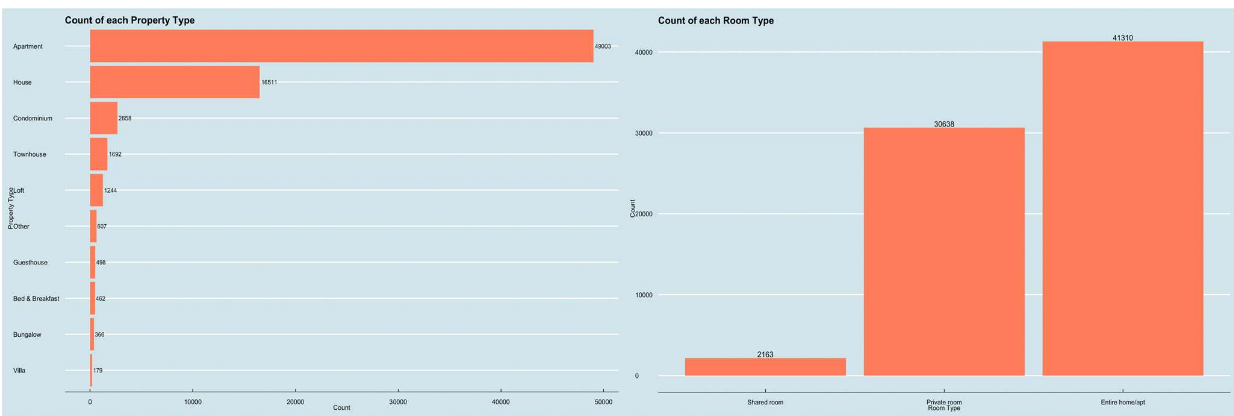


Figure 3a & 3b. Property Type and its Count

From the graph plotted in Figure 3a between Property type vs Count, the count of Apartment as stay is predominant followed by normal Houses with Villa being the least of all.

While analyzing for each Airbnb listing based on the general category, we were able to obtain the following plot (Figure 3b) where we could say that people prefer the entire home or apartment compared to the shared rooms.

The average price of different properties is listed in Figure 4. The property type Hostel is the one that is the cheapest and the property type Timeshare is the costliest.

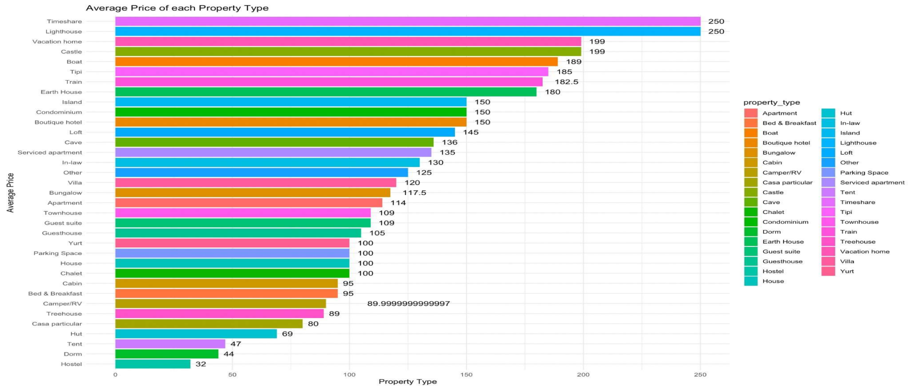


Figure 4. Average price of Property Type

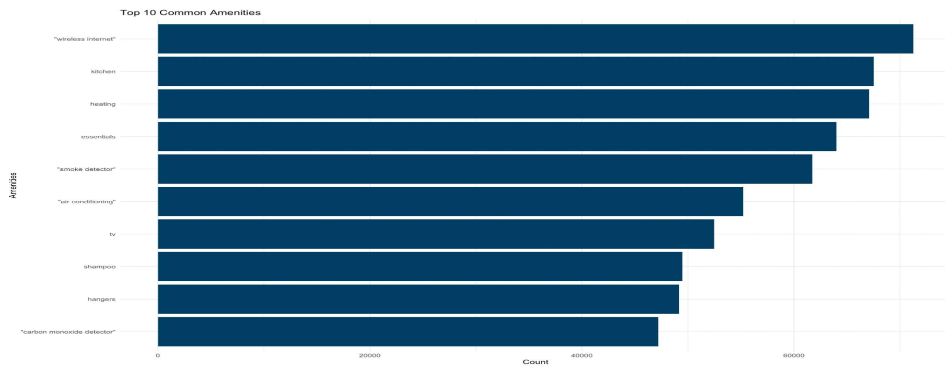


Figure 5. Top 10 Amenities

Wireless Internet (Wifi) is the common amenities across all the properties followed by Kitchen and Heaters. Almost 44000+ properties have all the above-mentioned amenities. (Figure 5)

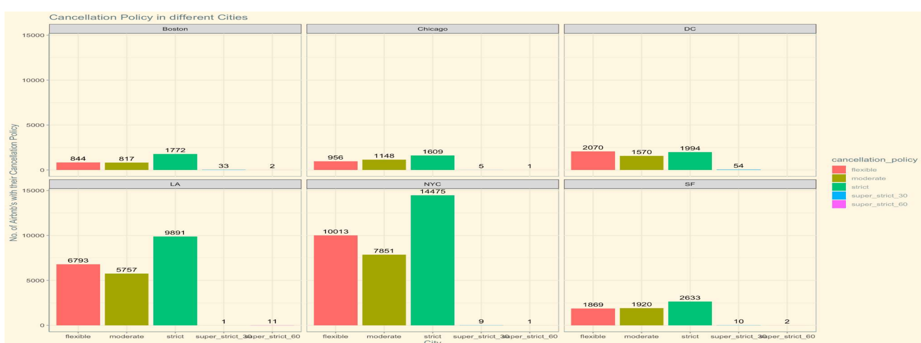


Figure 6. No. of Airbnb's with Cancellation policy in six Cities

The group of Bar graphs show the different cancellation policies followed in different properties in different cities. In Boston, cancellation policy seems not flexible, that is they have low flexibility and NYC Airbnb's having more properties with good flexibility for cancellation policy. The rest of the city listings can be observed above. (Figure 6)

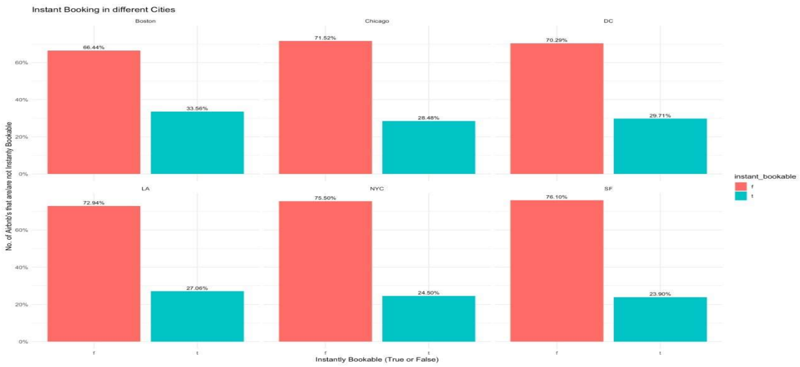
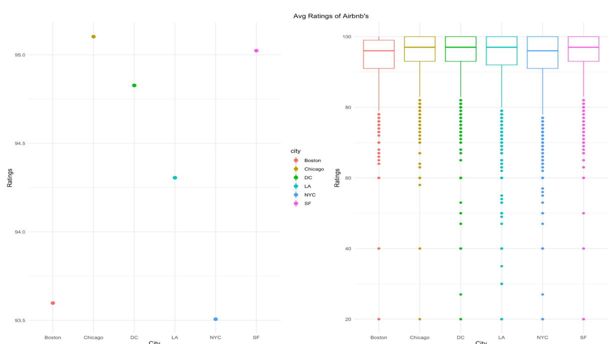


Figure 7. Instant Booking Option

Cities	Instant booking	Not Instant Booking
Boston	2304	1164
Chicago	2660	1059
DC	3998	1690
LA	16377	6076
NYC	24425	7924
San Francisco	4896	1638

As observed from the previous graphs, densely populated cities such as New York and Los Angeles have the premium option of instant booking and the ratio is equally managed between instant and non-instant booking subject to their high volume of listings. This can be clearly understood by observing Figure 7.



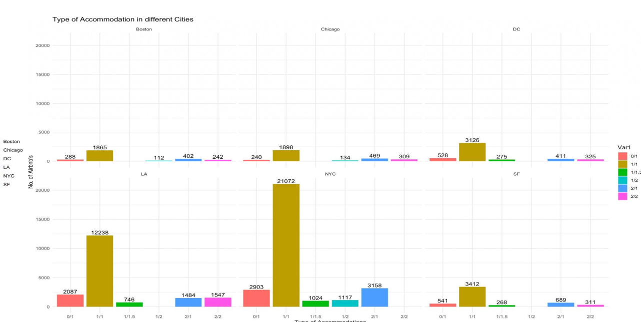


Figure 8. Average Ratings in different cities

Figure 9. Number of Airbnb's Vs type of accommodations

Chicago and San Francisco properties have the most positive reviews of above 95 ratings, whereas New York city is the least rated one with 93.5. For much more clarity we have also plotted the box plot for each individual city. (Figure 8)

Among all the cities, 1 bedroom with 1 bathroom is the most common type of accommodation/listing. (Figure 9)

The population density of various kinds of properties across the Map have been mapped and placed in Figure 10a and Figure 10b. It gives a brief idea about the various properties in the cities. Finding an exact price is not extremely important when surveying an Airbnb listing. A price range can be just as telling. So, we discretized the log price into six different categories that is from 0 to 1500 and plotted the density map for each city for which we got the following result.

The listing costs are largely in line with the location scores. Highly rated locations also tend to be the most expensive ones since the highly rated location would also tend to be costly.

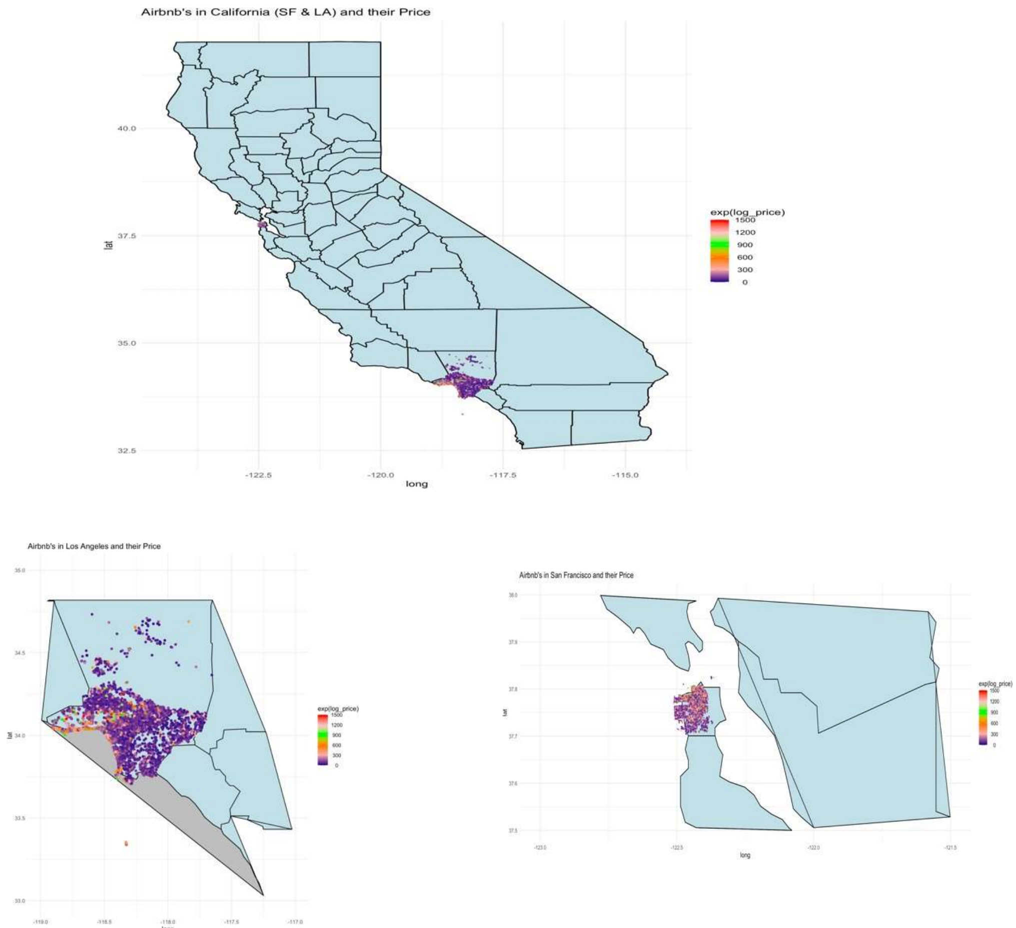


Figure 10. ***California (San Francisco and Los Angeles)***

Above 3 are the graphs for California state along with its cities San Francisco and Los Angeles.
We could observe that the price of a listing is expensive if it is closer to the bay area.

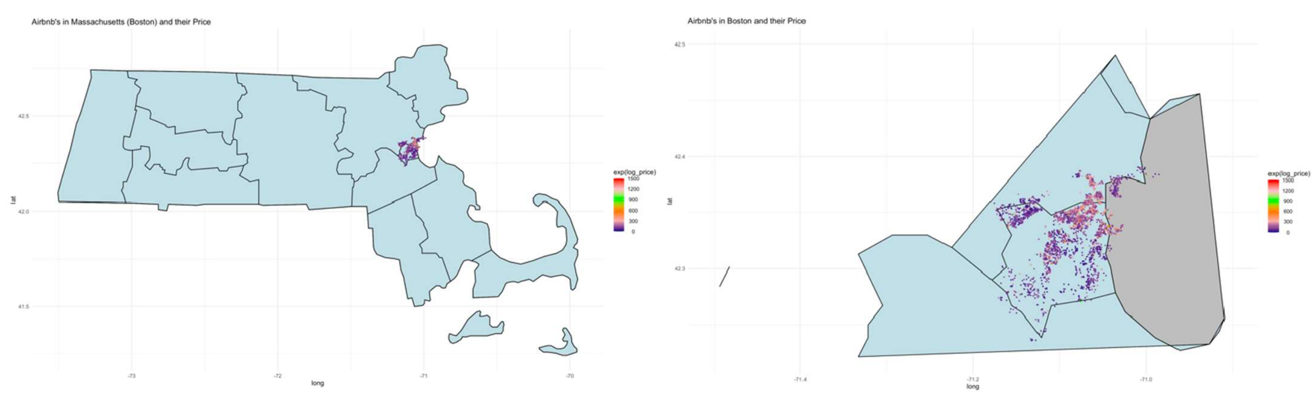


Figure 11. ***Massachusetts (Boston)***

Figure 11 displays the locations of the listings in Boston, Massachusetts. The same observations can be made here that closer the location to the ocean, Charles River and Boston downtown, higher is its price.

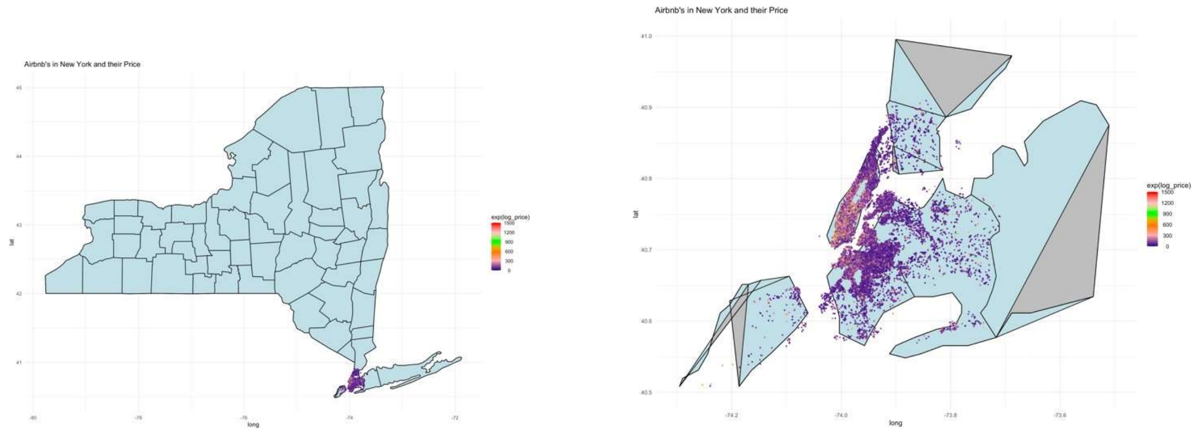


Figure 12. New York

In New York Properties along the Hudson River are the expensive when compared to other listings.

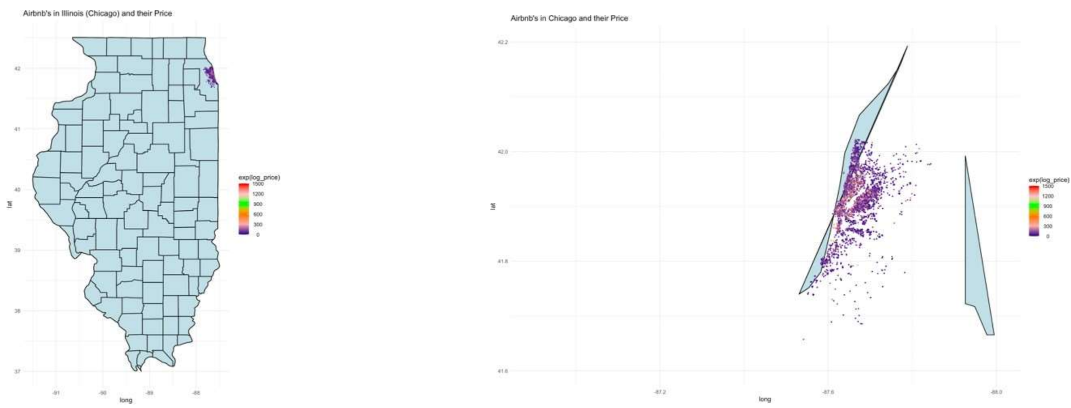


Figure 13. Illinois (Chicago)

From Figure 13 and 14 it would be clear that listings price changes with the change in its location. Prime located listings tend to have higher price when compared to others.

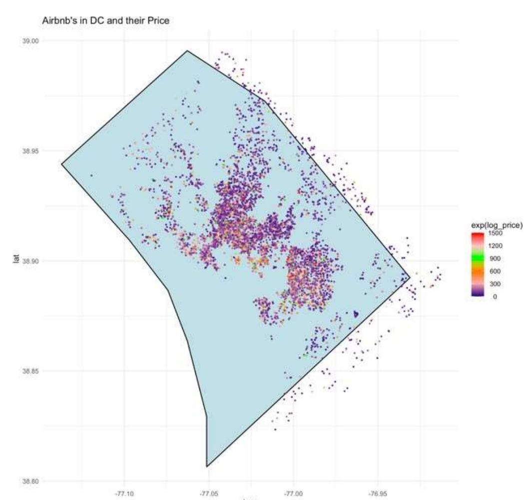


Figure 14. DC

Model	Train RMSE	Test RMSE
Linear Regression	0.43	0.44
Random Forest	0.40	0.42
SVM	0.41	0.42

Above are the results for the 3 models that have been implemented. Based on the results, the important factors that are to be considered while booking a property are:

- Room type
- Ratings
- Bedrooms
- Bathrooms

The most listed type of listings is “Apartments” and 1B1B type are more.

Properties Near Oceans, Rivers and Downtowns have high prices.

Also, the 10 common amenities in a listing are:

- Wireless Internet
- Kitchen
- Heating
- Essentials
- Smoke Detectors
- Air Conditioning
- TV
- Shampoo
- Hangers
- Carbon Monoxide Detectors

Discussions

From the results one can understand what people are looking in a listing before booking it. They would be considering room type, ratings, bedrooms, accommodations, and bathrooms. Moreover, from the analysis it is evident that price of the property may increase based on the location of it within the city. For example, in Los Angeles if a property is close to the bay area, the cost would be more when compared to a listing which is centrally located. This makes sense as most people would prefer the beach view and so is the demand for it.

This project will help people who want to post their listing on Airbnb. They will know what the customers are looking in a property, the amenities to be provided, importance of location and pricing.

Further, this project can be improved by gathering even more data across different cities and adding few other parameters like property area, its connectivity, etc. This would help is better analyzing the price.

Statement of Contributions

- Sai Venkata Manoj Vungarala: Data Collection, Data Transformation, Tokenization, Data Visualization, Exploratory Data Analysis, Step wise Modeling and Random Forest Regressor Model.
- Ishal Abhishek Mummidivarapu: General Trends analysis, Data Visualization, Linear Regression and SVM Modeling.
- Sujith Naarayan Hirendra Babu:. Preliminary Analysis, Data Collection, Cleaning and Preprocessing, Data Transformation, Data Visualization, Project Content and Information Collection and Preparation of Presentation.

For the report, everyone worked on the portion that they did the analysis and generated visualizations.

References

[1] Data from Kaggle (https://www.kaggle.com/rudymizrahi/airbnb-listings-in-major-us-cities-deloitte-ml)
[2] Text-Mining (https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/)
[3] Modeling (https://www.sciencedirect.com/topics/engineering/random-forest)

Appendix

```r
# Importing Libraries
library(tidyverse)
library(ggplot2)
library(tidytext)
library(dplyr)
library(ggthemes)
library(ggpubr)
library(modelr)
library(maps)
library(mapdata)
library(party)
library(curl)

# Importing the dataset
df <- read.csv("train.csv")
df$log_price <- exp(df$log_price)
names(df)[2] <- "price"

unique(df$city)

df$state <- ""
for (i in 1:nrow(df)){
  if ( df$city[i] == 'NYC') {
    df$state[i] <- 'new york'

  } else if ( df$city[i] == 'Boston') {
    df$state[i] <- 'massachusetts'

  } else if ( df$city[i] == 'LA' |  df$city[i] == 'SF') {
    df$state[i] <- 'california'

  } else if ( df$city[i] == 'DC') {
    df$state[i] <- 'district of columbia'

  } else if ( df$city[i] == 'Chicago') {
    df$state[i] <- 'illinois'

  }

}
```

O . A Go to file/function Addins •

Of IDMP Project.R '

Source on Save @ 4 Run ^4 Source

```
# The data
heod(df)

# Count of each Prope ray T\P" |
df  â>â  count  p roper ty_type ,   sort-TRUE    â>â
  top_n {10} @>%
  ggplot{aes{x-reo rder  proper ty_type , n}, y=n, l abel-n} +
  geom_col {ft Tl= ' #F18464 '}+
  coord_flip(j+
  geom_text(size=3.5,hjust=—0.1)+
  lobs(x='Property Type',y='Count',title="Count of eoch Property Type")*
  theme_econorrt st
  scole_ftll_economtst()

A Plotting the count of different Room Types

df %>% count(room_type, sort=TRUE) %>%
  top_n {10} @>%
  ggplot{aes{x-reorder roorr_type , ng , y-n, l abel -n g} +
  geom_col {ft Tl= ' #F18464 ' }+
  geom_text(size=4.5,vjust=—0.2]+
  Iabs {x= ' Room Type ' , y- ' Count ' , ttU e-" Count of each Room Type"}+
  theme_econorrt st
  scole_ftll_economtst()

# Pl ott i ng t he ave ra ge pr I c e of e ach p rop e r ty typ e
df %>%
  group_by{property_type â>â
  summari se{MedPrIce-    medi an{pr Ice,  no . rrr    TRUE    %>â
  ggplot(oes(x = reorder(property_type,MedPrice), y = MedPrice,lobel=MedPrice,fill=property_typej]
  geom_col()+
  coord_flip(j+
  geom_text(size=4.0,hjust=—0.5)+
  lobs(x="Averoge Price", y= "Property Type", tttle="Averoge Price of eoch Property Type")+
  theme_minimal()


# Plotti ng the most con1non Anenttes
df$omenttes <- substr(df$omenttes, stort=2, stop=nchor(df$omenitiesj-1)
df_tz <  unnest_to kens df , Ameni ti es , amenl li  es ,  to ken-   ' reg ex ' ,  pattern-" , "

df_tz Z >%
  count{Amen II Ies,  sort=TRUE   @>%
  top_n{10} @>%
  ggplot{aes{x-reorder{AmenII I es , n}, y-n} +
  geom_col(ftll='#023E62')
```

O • | | • | @ | | Go to file/function %o Addins •

IDMP Project.R* x

| | @ Q Source on Save | @ - • | Q I-6 Run | I^6 | Source

```
  coord_ft I pC$ +
  t absCx="Amenities " , y="Count" ,
       titte="Top 10 Common AmeniU es "} +
  theme minimol(J

# Cancel tation Policy I n the six Cities
ggplot(df,oes(x=cancellotion_policy,fill=cancellotion_policy)J+
  geom_bor(J+
  focet wrop(-cityJ+
  geom text(stat='count', oes(lobel=..count..), vjust=-0.5)+

  tabsCx = ' City' , y = "No . of Ai rbnb ' s with thel r Cancel tation Posi cy" ,
       title = "Cancel tation Pot icy i n dt fferent Cities " )+
  theme sotarizedC}

#I nstant Booking I n the six Cities
df %>% gnoup_by(city) %>%
  count( instant_bookoble = facton(instant_bookQbleJ) %>X
  mutote(pct = prop.toble(nJJ%>%
  ggplot(aes(x=instant_bookQble, y= pct, fill = instant_bookable, lobel = scoles::pencent(pct)JJ +
  geom_bor(position = 'dodge',stat='identity') +
  focet wrop(-cityJ+
  geom text(position = position_dodge(width = .9J,
            vjust = -0.5,
            size = 3) +
  scate_y_conti nuous{labets = scotes : :percent)+
  tabsCx = ' Instantly Bookabl e (True or False} ' , y = " No . of At rbnb's that are/ore not Instonty B
       title = "Instant Bookt ng in di fferent Cities " }+
  theme mini mat {}


# No. of Ainbnb's
ggplot(df,oes(x=os.fOctor(city),fill=cityJJ+
  geom_bor(J+
  geom text(stat='count', oes(lobel=..count..), vjust=-0.5)+
  lobs(x = "City",y = "Count of Airbnb's", title = "No. of Ainbnb's")+
  theme economist(J +
  scat e_fi11 econooi st()

# Avg Ratings of Airbnb ' s

plotl <- ggplot(df,aes(x=city,y=review_scores nating,color=city)J+
  stat summary(fun="meo n" }+
  tabsCx="City" ,y="Rotings"} +
  theme minimol(J
```

O • | | • | @ @ | | Go to file/function %o Addins •

IDMP Project.R* x

| | @ Q Source on Save | @ - • | Q |-G Ru n | |••G | Source

```
plot2 <- ggplot(df,aes(x=city,y=review_scores nating,color=city)J+
  geom_boxptot()+
  tabsCx="City" , y="Rotings"} +
  theme mini mat { }

Fig <- ggarrongeCpt ot1,ptot2}
annotate ft gure{ft g, top = text grobC "c ")}

# Bedrooms/Bathrooms


ddf<- df %>% mutote(concated_column = poste(bedrooms, bQthnooms, sep = '/'JJ
ddf<-os.dato.frome(toble(ddf#concoted_column,ddf#cityJJ  %>%
  group_by{Vor2} B>S
  sti ce max{order_by = Freq , n = SQ B>B
  group_by{Vor2} B>S st l ce maxCorder_by = Freq , n = 5)

ggpt ot Cddf, oesCx = Var1, y = Freq , ftTt=Var1}g +
  geom_bar{stat = " i dentity" } +
  geom text Coes{labet = Freq} , vj ust = -0.3) +
  facet wrapC-Vor2}+
  lobs(x = 'Type of Accommodotions', y = " No. of Airbnb's",
       title = "Type of Accommodotion in different Cities" J+
  theme_minimal()

# Maps

mop_df <- mop_dota("stote"J
mop_dota <- left join(mop df, df, by = c("region"="stote"J)

states <- map_dato{ " state"}
countt es <- map dota( "county"}

# Colifonnio Mop
co df <- subset(stotes, region == "colifornio"J
co county <- subset(counties, negion == "colifornio"J
city_co <- map doto("stote",negion='colifonnio')
df_ca <- filter(df,df$stote =='colifonnio'J

co_base <- ggplot(dato = co df, mopping = aes(x = long, y = lOt, gnoup = gnoupJ) +
  coond_fixed(1.3) +
  geom polygon(color = " bl ack " , ft ll = " gray ")

ca_base <— ggptotCdata = ca df, mapping = aes{x = long, y = tot, group = group}) +
  coord_fixedC1.3) +
  geom polygon(color = "bl ack " , VIII = "gray" )
```

Go to file/function Add ins

Of IDMP Project.R*

Source on Save @ '- 4 Run °4 Source

```
p<—c(0,300,600,900,1200,1S00)

co_bose
  geom_polygon(doto = co_county,fill='#C7DEE5',color  "block") +
  geom_polygon(color  "block", fill = NA)+
  geom_potnt(doto=df_co,oes(x=longitude, y=lotitude,color=prtce),tnhertt.oes = FALSE,stze=0.1)*
  lobs(title = "Atrbnb's in Colifornio (SF & LA) ond their Price"j*
  scale_color_gradientn{l imt ts    c{0,1500},
                              colours-c{"novybl ue", "#E EB1AA", "darkorange1",'green','pi nk' ,"red" ,
                              breaks=p , labels=format{p}}+
  theme_mt nimal{g

co_bose
  geom_polygon(doto = co_county,fill='#C7DEE5', color   "block") +
  geom_polygon(color  "block", fill = NA)+
  geom_potnt(doto=df_co,oes(x=longitude, y=lotitude,color=prtce),tnhertt.oes = FALSE,stze=0.5)*
  xli m(—123,— l2l .5) › yli in(37 . 5, 38)•
  scale_color_g radientn{l imtl s    c{0,1500},
                              col ours-c "novybl ue", "#EEB1AA", "dar korange1",' gre en','pi nk'," r ed" ,
                              b rea ks=p , l abel s=format p} } +
  labs{titl  e    "At rbnb's I n San Franct sco and the I r Prt ce" }+
  theme_ml ni mat

co_bose
  geom_polygon(doto = co_county,fill='#C7DEE5', color   "block") +
  geom_polygon(color = "block", fill = NA)+
  geom_potnt(doto=df_co,oes(x=longitude, y=lotitude,color=prtce),tnhertt.oes = FALSE,stze=0.5)*
  xl tm{—119,— 117} + yt tu(33, 35g+
  scale_color_g radientn{l imtl s    c{0,1500},
                              col ours-c "novybl ue", "#EEB1AA", "dar korange1",' gre en','pi nk'," r ed" ,
                              b rea ks=p , l abel s=format p} } +
  labs{titl  e    " At rbnb's I n Los Ang el es and I hel r P r l ce   +
  theme_ml ni mat

A Boston MQp
mo_df <— subset(stotes, regton == "mossochusetts")
mo_county < subset(counties, region == "mossochusetts")
city_mo<  mop_doto("stOte",region='mossochusetts'j
df_mo <— ftlter(df,df$stote == 'mossochusetts']

ma_base <  ggplot data   ma_df, mapping   aes{x   l ong, y   l at , group   group } } +
  coord_ftxed{1.3} +
  geom_pol ygon colo r — "bl ack" , fi Tl -   " gray "}

ma  base <  oaolot Ldata   ma df . ffIODDino   aesix   l ono . v   l at . arou a   arouo   +
```

a - - @ Go to file/function % Addins •

IDMP Project.R"

@ Q Source on Save @ '- • /l I-6 Run I^6 Source

```
ma_bose <— ggptot{dota = ma_df, mapping = oes{x = tong , y = tot, group = group)} +
 coord_fixedC1.3} +
    geom polygon(color = " bl ack " , ft II = " gray " }

mo_bose <- ggplot(doto = mo_df, mopping = oes(x = long, y = lot, gnoup = gnoup)) +
 coord_fixedC1.3} +
 geom polygon(color = " bl ack " , ft II = " gray " }

ma_bose +
 geom polygon(data = ma_county , VIII= ' DC 7DEEN ' , color = "bl ack " } +
 geom polygon(color = "bl ack " , VIII = NA}+
 geom poi nt{dota=df_mo , oes{x=tongitude, y=tatttude , cotor=price} , i nherit .oes = FALSE , size=0. 1}+
 tabsCtt tt e = "Ai rbnb' s in Massachusetts CBostong and their Price " }+
 scote cotor_gradientnCt lnnits = c(0,1500},
                    cot ours=cC" navybl ue" , "#EEB1AA" , " dorkoronge1" , 'green' , 'pink' , "red"} ,
                    breoks=p , tabets=format{p)}+
 theme mini mat { }


ma_bose +
 geom polygon(data = ma_county , VIII= ' DC 7DEEN ' , color = "bl ack " } +
 geom polygon(color = "bl ack " , VIII = NA}+
 geom poi nt{dota=df_mo , oes{x=tongitude, y=tatttude , cotor=price} , i nherit .oes = FALSE , size=0.5}+
 xl im{ -71. g, —70. 9) + yl inn(42.2Z , 4Z . 5g+
 tabsCtt tt e = "Ai rbnb' s in Boston and their Price"}+
 scote cotor_gradientnCt lnnits = c(0,1500},
                    cot ours=cC" navybl ue" , "#EEB1AA" , " dorkoronge1" , 'green' , 'pink' , "red"} ,
                    breoks=p , tabets=format{p)}+
 theme mini mat{ }

# New Yo r k Map
ny_df <- subset{states , region == "new york"}
ny_county <- subset(counties, region == " new york"}
city_ny <— mop_data( " state", region= ' new york' }
df_ny <- fi l ter{df ,df$state == ' new york' }

ny_bose <- ggplot(doto = ny_df, mopping = oes(x = long, y = lot, gnoup = gnoup)) +
 coord_fixedC1.3} +
 geom polygon(color = " bl ack " , ft II = " gray " }

ny_bose <- ggplot(doto = ny_df, mopping = oes(x = long, y = lot, gnoup = gnoup)) +
 coord_fixedC1.3} +
 geom polygon(color = " bl ack " , ft II = " gray " }

ny_bose +
 geom polygon(data = ny_county, VIII= 'DC 7DEE5 ' , color = "bl ack "} +
```

a - - @ Go to file/function % Addins •

IDMP Project.R*

Q Q Source on Save @ J*" /| I-4 Run I^4 Source

```
ny_bose +
  geom polygon(data = ny_county, VIII= 'DC 7DEE5 ' , color = "bl ack "} +
  geom polygon(color = "bl ack " , VIII = NA}+
  geom  poi  nt{dota=df_ny,oes{x=tongitude, y=tatttude , cotor=price} , i nherit .oes  =  FALSE , size=0. 1}+
  scote cotor_gradi entnCtlnnits     = c(0,1500},
                                cot ours=cC "navybl ue" , "#EEB1AA" , "dorkor onge1" , 'green' , 'pink' , "red"} ,
                                breoks=p ,  tabets=format{p)}+
  tabsCtt tt e = "Ai rbnb' s in New York and their Price" } +
  theme mini mat { }

ny_bose +
  geom polygon(data = ny_county, VIII= 'DC 7DEE5 ' , color = "bl ack "} +
  geom polygon(color = "bl ack " , VIII = NA}+
  geom poi nt{dota=df_ny, oes{x=tongitude, y=tatttude , cotor=price} , i nherit .oes = FALSE , size=0.5}+
  xlim{-74.3, —73.5) + yl inn(40.5, 41}+
  scote cotor_gradi entnCtlnnits     = c(0,1500},
                                cot ours=cC "navybl ue" , "#EEB1AA" , "dorkor onge1" , 'green' , 'pink' , "red"} ,
                                breoks=p , tabets=format{p)}+
  tabsCtt tt e = "Ai rbnb' s in New York and their  Price"
  theme mini mat { }

# Chl cago  Map
chi_df <- subset C states , region == "i ll i noi s"}
chi_county <- subset{counti es , region == "iU inois"}
ci ty_chi <- map_dot o{ " state" , region= ' it l i noi s ' }
df cht <- fi tterCdf, dfsstote == ' i l l i noi s' )

chi_bose <- ggplot(doto = chi_df, mopping = oes(x = long, y = lot, group = gnoupJ) +
  coond_fixed(1.3) +
  geom polygon(color = "block", fill = "gnoy")

chi_bose <- ggplot(doto = chi_df, mopping = oes(x = long, y = lot, group = gnoupJ) +
  coond_fixed(1.3) +
  geom polygon(color = "block", fill = "gnoy")

chi_base +
  geom polygon(doto = chi county,fill='#C7DEE5', colon = "block"J +
  geom polygon(color = "bl ack " , VIII = NA}+
  geom  poi  nt{dota=df_cht , oes{x=tongitude ,  y=lotitude , cotor=prt ce} , I nhert t . aes =  FALSE , size=0. 1}+
  scote cotor_gradi entnCtlnnits     = c(0,1500},
                                cot ours=cC "navybl ue" , "#EEB1AA" , "dorkor onge1" , 'green' , 'pink' , "red"} ,
                                breoks=p ,  tabets=format{p)}+
  tabsCtt tt e  =  "Ai rbnb' s  in   IIIinoiCChicago}  and  their  Price"}+
  theme mini mat { }
```

Go to file/function Addins •

Of IDMP Project.R'

Source on Save @ @ 4 Run ^4 Source

```
chi _base +
  geom_polygon(doto = cht_county,ftll='SC7DEE5', color   "block"]
  geom_polygon(color   "block", fill = NA)+
  geom_potnt(doto=df_chi,oes(x=longttude, y=lotttude,color=price],inherit.oes =FALSE,size=0.5)+
  xltm( 86.9, 88)  yltm(41.6, 42.2)
  lobs(title="ChicOgo")+
  scab e_color_gradientn{lirrt ts    c{0,1500},
                        colours-c{"navybl ue", "#EEB1AA", "darkoranpe1","green','pi nk' ,"red" ,
                        breaks=p , labels=format{p}}+
  lobs(title = "Atrbnb's in Chicogo Ond their Price")+
  theme_rrtnimal{g

A DC MOp
dc_df <- subset(stotes, regton    "district of
dc_county < subset(counttes, region  = "district of columbto")
city_dc <- mop_doto("stote",region='district of columbto')
df_dc<  ftlter(df,df$stote == 'district

dc_bose<  ggplot(doto = dc_df, mopping = oes(x   long, y  lot, group = group)) +
  coord_ftxed(1.3) +
  geom_polygon{color   "black",  fill    "gray"

dc_bose<  ggplot(doto = dc_df, mopping = oes(x   long, y  lot, group = group)) +
  coord_ftxed(1.3) +
  geom_polygon(color   "block", fill   "groy")

dc_bose *
  geom_polygon(doto = dc_county,fill='#C7DEE5', color   "block")
  geom_polygon{color   "black",  fist-   NA}+
  geom_potnt(doto=df_dc,oes(x=longitude, y=lotitude,color=prtce),tnhertt.oes = FALSE,stze=0.5)+
  scab e_color_gradientn{lirrt ts    c{0,1500},
                        colours-c{"navybl ue", "#EEB1AA", "darkoranpe1","green','pi nk' ,"red" ,
                        breaks=p , labels=format{p}}+
  lobs(title = "Atrbnb's in DC Ondtheir Price")+
  theme_mtnimal{g

# Model
set . s eed{3
df_part < r esampl e_pa rt l li on df ,
                              p-c{trat n-0.6,
                                vol l d-0.2,
                                test-0.2}
```

Go to file/function Addins •

Of IDMP Proj ect.R '

Source on Save @ • 4 Run ^4 Source

```
stepl <- functton(response, predictors, condtdotes, portition]
(
  rhs <- poste0(poste0(predtctors, collopse="+"), " ", condidotes)
  formulos <- lopply(poste0(response, "-", rhs), os.formulo]
  rmses <- sopply(formulos,
                 function(fm) rmse(lm(fm, doto=portttion$troinj,
                                    doto=portittonBvoltd))
  nomes(rmses] <  condtdotes
  ottr(rmses,       <   rmses#which.min(rmses]j
  rmses
1
model <- NULL

# Step-1
preds <— " 1 "
conds <- c("property_type","room_type", "Occommodotes", "bathrooms",
           " ci ty " , " be drooms " , " review_s core s_ratl ng "
s1 -< step1{ "l og_p r l ce " , pred s , conds , dE_pa rt

model <- c{model , att r s1, "best "
s1

# St ep-Z
pr eds <  " room_type"
conds <- c("property_type","Occommodotes", "bothrooms",
          "city", "bedrooms", "review_scores_rottng"]
s1 -< step1{ "l og_p r l ce " , pred s , conds , dE_pa rt

model <- c{model , att r s1, "best "
s1

# St ep-3
preds<   c("room_type", "review_scores_roting"]
conds <- c("property_type","Occommodotes", "bothrooms",
           "city", "bedrooms")
s1 -< step1{ "l og_p r l ce " , pred s , conds , dE_pa rt

model <- c{model , att r s1, "best "
s1

# Step —^
preds<   c("room_type", "review_scores_roting", "bedrooms"j
conds <- c("property_type","Occommodotes", "bothrooms",

s1 <- step1{ "l og_pr l ce " , preds , conds , d f_part
```

Go to file/function

Addins •

Of IDMP Project.R'

Source on Save @ -

4 Run ^4 Source

```
model <  c{model , attr s1, "best"
s1
# Step —^
preds <— c("room_type", "review_scores_roting", "bedrooms","city")
conds <— c("property_type",”Occommodotes”, "bothrooms")
sl <- step1(”log_prtce”, preds, conds,df_port]
model <  c{model , attr s1, "best"
s1
# Step 5
preds<  c("room_type", "review_scores_roting", "bedrooms","city","occommodotes")
conds <— c("property_type",”bothrooms”j
sl <- step1(”log_prtce”, preds, conds,df_port]
model <  c{model , attr s1, "best"
s1
step_model <  tibble{index-seq_along{ model } ,
                    variable-factor{names{model , levels-names{ model } },
                    RMSE=model}
ggplot step_model , aes{y-RMSE}} +
  geom_pot nt aes{x=var I able}
  geom_li ne{ aes{x-index} } +
  I abs{titl e-" Stepwt se model set ection "} +
  theme_minimal()
fit_model < lm(dg  prtde - room_type * review_scores_rottng * bedrooms + ctty ,
ftt5 < lm{log_pr I ce - roorr_type + review_scores_rat I ng + bedrooms + ctty ,
          doto = df_portBtrotn]
rmse(fit5,df_port5troinj
rmse(fit5,df_port5volid)
rmse(fit5,df_port5test)
# Random Forest
rf <- rondomForest( og_price - room_type   review_scores_rottng   bedrooms   ctty,
                    doto = df_port$trotn, mtry=3,importonce = TRUE, no.octton = no.omits
prtnt(rfJ
```

```r
print(rf)

# Step 1 - predicting and evaluating the model on train data
train_pred = predict(rf, newdata = df_part$train)
rmse(rf, df_part$train)

# Step 2 - predicting and evaluating the model on test data
test_pred = predict(rf, newdata = df_part$test)
rmse(rf, df_part$test)

# SVM
library(e1071)

msvm <- svm(log_price ~ bedrooms+city+room_type+review_scores_rating, data = df_part$train)

# Step 1 - predicting and evaluating the model on train data
train_pred = predict(msvm, newdata = df_part$train)
rmse(msvm, df_part$train)

# Step 2 - predicting and evaluating the model on test data
test_pred = predict(msvm, newdata = df_part$test)
rmse(msvm, df_part$test)
```

For Linear Regression

```
R 4.1.1 · ~/Desktop/IDMP/Project/
> rmse(fit5,df_part$train)
[1] 0.4392617
> rmse(fit5,df_part$valid)
[1] 0.4339424
> rmse(fit5,df_part$test)
[1] 0.440563
```

For SVM

```r
library(e1071)

msvm <- svm(log_price ~ bedrooms+city+room_type+review_scores_rating, data = train)

# Step 2 - predicting and evaluating the model on train data
train_pred = predict(msvm, newdata = train)
rmse(msvm, train)

# Step 3 - predicting and evaluating the model on test data
test_pred = predict(msvm, newdata = test)
rmse(msvm, test)
```

```
[1] 0.4150172
[1] 0.4254698
```

For Random Forest

```
```{r}
# Load the library
library(randomForest)

train

rf <- randomForest(log_price ~ bedrooms+city+room_type+review_scores_rating, data = train, mtry=3,importance = TRUE, na.action = na.omit)

#print(rf)

# Step 2 - predicting and evaluating the model on train data
train_pred = predict(rf, newdata = train)
rmse(rf, train)

# Step 3 - predicting and evaluating the model on test data
test_pred = predict(rf, newdata = test)
rmse(rf, test)

```
```

data.frame

R Console

```
[1] 0.4051719
[1] 0.4266298
```